



Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcomEuQoS: End-to-End Quality of Service over Heterogeneous Networks [☆]

E. Mingozzi ^{a,*}, G. Stea ^a, M.A. Callejo-Rodríguez ^b, J. Enríquez-Gabeiras ^b, G. García-de-Blas ^b, F.J. Ramón-Salquero ^b, W. Burakowski ^c, A. Beben ^c, J. Sliwinski ^c, H. Tarasiuk ^c, O. Dugeon ^d, M. Diaz ^e, L. Baresse ^f, E. Monteiro ^g

^a Dipartimento di Ingegneria dell'Informazione, University of Pisa, Via Diotisalvi 2, 56122, Pisa, Italy

^b Telefónica I+D, Madrid, Spain

^c Warsaw University of Technology, Warsaw, Poland

^d France Telecom R&D, Lannion, France

^e Université de Toulouse/LAAS-CNRS, Toulouse, France

^f Silogic, Toulouse, France

^g CISUC/DEI, University of Coimbra, Portugal

ARTICLE INFO

Available online xxxx

Keywords:

EuQoS

Quality of Service

Network Architecture

Heterogeneity

ABSTRACT

The EuQoS (*End-to-End QoS over Heterogeneous Networks*) IST Integrated European Project aimed to define a Next Generation Network architecture that builds, uses and manages end-to-end QoS across different administrative domains and heterogeneous networks (UMTS, xDSL, Ethernet, WiFi, Satellite and IP/MPLS). The EuQoS architecture preserves the openness and the decentralized decision model of the actual Internet, runs on off-the-shelf hardware and network equipment, and allows end users to request various services without changing the Application Signaling protocol, while meeting regulators' and users' Net Neutrality requirements. This paper presents the key elements of the EuQoS architecture and describes the main results obtained in field trials performed on a fully-functional EuQoS system prototype developed over a pan-European testbed. Furthermore, the paper discusses the main strengths of the system and the issues related to its actual deployment on a large scale, from both technical and market points of view.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

The user demand for multimedia application over the Internet has been rapidly growing in the past few years. Voice over IP (VoIP), and especially IPTV and Video on Demand (VoD) applications are gaining an ever increasing popularity [33], favored by the massive deployment of diverse access technologies, such as 802.11/WiFi, UMTS (Universal Mobile Telecommunications System), xDSL (variants of Digital Subscriber Line), and Ethernet LANs. Supporting these applications requires Quality of Service (QoS) provisioning and management at all the relevant points in the Internet. In particular, QoS provisioning implies to master the cooperation of several building blocks (e.g., routing algorithms, resource management schemes, admission control algorithms, traffic analysis techniques, signaling protocols). While each of the above topics has been the subject of extensive research in the last fifteen years, as testified by the abundance of related literature, up to now

only partial solutions, addressing either specific network technologies or specific building blocks have actually been deployed. Among the reasons that have prevented deployment, there are the well-known problems of *network heterogeneity*, and *decentralized control* of the Internet, made very complex by the needed properties of *scalability* and *reliability* of QoS provisioning. On one hand, QoS provisioning is effective only if it is achieved on a full end-to-end basis. As stated in [13], an architecture designed to address this goal should integrate and synchronize various tasks performed in the different planes and network segments of the end-to-end path. However, the Internet is composed of loosely coupled Autonomous Systems, whose policies are largely independent: while several techniques exist that address QoS provisioning *inside* a single domain, solutions that allow domains to setup compatible QoS policies are generally lacking. Second, the various network technologies that compose an end-to-end path have different capabilities in terms of bandwidth, delay, and forwarding capabilities, which makes its design impossible using one unified solutions for all technologies. Third, QoS policies have to cope with a potentially large number of users. Fine-grained solutions, such as complex traffic classification or per-flow packet queuing, have already been proved to hamper scalability. Last, but certainly not

[☆] A preliminary version of this paper has appeared in *Proceedings of K-INGN 2008, Geneva (CH), May 12–13, 2008*.

* Corresponding author. Tel.: +39 050 2217 655.

E-mail address: e.mingozzi@iet.unipi.it (E. Mingozzi).

least, QoS is expected to generate new revenue for the network operators. However, it seems that no customer is ready to pay for an unreliable and not certifiable level of QoS.

In the recent past, several fora have addressed the problem of setting up a large-scale QoS architecture, without however taking into account some of the above mentioned aspects. Recent European projects, like TEQUILA [17], MESCAL [18] and AGAVE [48] were aimed at establishing Inter-Domain Traffic Engineering policies (e.g., Service Level Specifications) to allow QoS to be negotiated beyond the borders of single domains. However, they explicitly focused on IP domains using DiffServ and/or MPLS (i.e., on the backbone), not taking into account the (possibly heterogeneous) access segments of the end-to-end path. The DAIDALOS project [49], has studied the problem of providing pervasive and user-centered access to services over diverse network technologies beyond 3G. As such, it has mainly tackled the (wireless) access segments of the end-to-end path (802.11e, 802.16, 802.15.1, TD-CDMA). While integration of access segments with a multidomain core is considered, large-scale QoS provisioning issues, such as QoS routing and traffic engineering in the core domains, are given a comparatively minor attention. The US-based QBone project [19] of the Internet2 consortium has addressed the problem of deploying scalable QoS in the Internet, ending up with proposing two modified versions of the classical best-effort service and advising against the deployment of “premium” services in the Internet *tout-court*. Perhaps the most thorough effort in that sense has been the IP Multimedia Subsystem (IMS) [26], developed within the framework of the Third Generation Partnership Project (3GPP). The main inconvenience of IMS is the *mandatory* requirement of the Session Initiation Protocol (SIP, [11]) to interact with the main coordination point, i.e., the Proxy Call Session Control Function (P-CSCF). This does not allow applications using different signaling protocols to use the IMS framework. Furthermore, even though the current IMS architecture is designed to interact with the ETSI/TISPAN Next Generation Network (NGN) framework, a solution to provide QoS valid for heterogeneous network technologies and able to coordinate the QoS mechanisms available in the different domains has neither been defined nor implemented yet. Finally, the IPSphere forum [27] takes care of specifying the relationships (by means of interfaces) among the different Internet stakeholders (Internet Service Providers, Content Providers, Internet carriers, etc.). The IPSphere framework addresses service negotiation and composition, relying on an underlying QoS framework. Specifically, it plans to use the NGN architectures being defined in the IMS and ETSI/TISPAN fora.

To the best of our knowledge, the first global user-to-user architecture addressing the problem of *end-to-end* QoS over *heterogeneous* networks has been devised in the framework of the IST-EuQoS project [1] a joint effort of European universities, research centers and major telecom operators. The *EuQoS system* is in fact able to satisfy QoS requirements (i) on an end-to-end basis; (ii) in a heterogeneous network scenario; (iii) at all relevant layers, ranging from the service to the network control and data planes; (iv) at all relevant timescales, ranging from network planning to packet forwarding; (v) being scalable to large dimensions, in terms of number of users and of involved Autonomous Systems; and (vi) relying on standard Internet protocols as much as possible, enhanced with new functionalities when needed. The main goal of the design is to support the evolution of the Internet into a multi-service network, taking a pragmatic approach that preserves the Internet openness. The *EuQoS system* encompasses both the core IP network and the currently most popular access technologies, such as UMTS, WiFi, xDSL and LAN/Ethernet. It specifies a set of end-to-end *network Classes of Services* (CoSs), which are then used to support a broad range of applications, such as VoIP, VoD, Tele Engineering, Distance

Learning, and Telemedicine, besides standard TCP-based applications. A prototype of the *EuQoS system* has been deployed into a pan-European testbed, consisting of twelve fully functional testbeds located in various European countries, connected to the GÉANT backbone [37] through their national research networks (NRENs).

The strength of the *EuQoS architecture* lies in the fact that it does not require current applications to be modified in order to reap the benefits of guaranteed QoS (although new applications can in fact be deployed so as to be *EuQoS-aware*); *EuQoS* allows the end users to request a specific QoS-guaranteed connectivity, independently of the chosen applications, thus meeting Net Neutrality requirements. Furthermore, it does not rely on specialized hardware, nor on modifications in the existing network equipment: the software modules lying at the core of the *EuQoS system* run on off-the-shelf PCs, and they interact with commercial routing equipment, configuring the latter in order to achieve the desired QoS objectives. This way, new hardware and possibly entirely new network technologies can be easily integrated into the *EuQoS framework* by simply writing new software drivers. Finally, the *EuQoS system* does not alter the decentralized Internet model, as it still relies on bi-lateral agreements between neighboring domains.

This paper describes the *EuQoS system*. The design, implementation and validation of the *EuQoS system* has involved about one hundred people for a 3.5 year period. As a result, the number of considered sub-problems, and the technical depth of the devised solution to each of those, are such as to make it impossible to attempt an omni-comprehensive description. The interested reader can find plenty of material in the project deliverables [1] and in the numerous papers published by the project partners. For this reason, this paper focuses on how the various building blocks of the architecture (e.g., signaling, routing, provisioning, admission control, monitoring, etc.) are combined into a unified framework. Furthermore, we analyze in detail how heterogeneity is managed within *EuQoS*. We report the results of field tests on the final version of the *EuQoS prototype*, showing proof of concept that end-to-end QoS is actually achievable, and discuss the implications of extending the system to larger scales and deploying it in the actual Internet.

The rest of the paper is structured as follows. Section 2 presents the *EuQoS architecture* and the processes involved in the provisioning of end-to-end QoS. Section 3 presents the *EuQoS prototype testbed* and reports results of field trials. In Section 4, we discuss the main points of strength of the *EuQoS system*, and the issues involved in deploying the *EuQoS system* on a large scale in the actual Internet. We report conclusive remarks in Section 5.

2. The *EuQoS system*

The natural starting point for understanding the *EuQoS system* is the description of its *QoS model*, explaining the fundamental architectural choices. After that, we describe how a user application requests a QoS-guaranteed connection to the *EuQoS system*. We then change the perspective, examining the system from a network-centric point of view and describing the actions that take place in the network to:

- provision resources in the network for QoS-guaranteed connections (*provisioning process*),
- reserve the provisioned resources, binding them to incoming connections (*invocation process*),
- control the status and the QoS of ongoing connections (*Operation, Administration & Maintenance process, OAM*), detecting and correcting QoS disruptions and system malfunctioning.

2.1. The QoS model

The EuQoS approach for composing QoS taking into account network heterogeneity is to provide a common framework, abstracting from network-specific details. Since QoS must be provided on an end-to-end basis, a consistent view of CoSs and related QoS requirements is needed in the whole network. For this purpose, *end-to-end* CoSs are implemented in EuQoS following IETF recommendations [7]. The submitted traffics are marked accordingly (by the applications and/or the ingress node). These CoSs (e.g., *Telephony*, *Multimedia Streaming*, etc., also shown in Table 1) specify the target QoS requirements on end-to-end delay, jitter, and packet loss rate [8,9] (henceforth collectively referred to as the *QoS parameters*). A best-effort, *Standard* class is also defined for non-QoS traffic.

In order to provide guaranteed QoS across heterogeneous network technologies, we need to define a way to implement a consistent forwarding treatment for traffic belonging to each CoS into each network technology. The current network technologies already allow, in some cases, a certain degree of differentiated forwarding treatment among different types of traffics, appropriately marked. For instance, in the 802.1p standard, a field in the MAC header specifies eight priority levels; the WiFi Wireless Multi Media (WMM) standard allows one to define Enhanced Distributed Control Access (EDCA) Access Categories (AC), which compete for medium access using different contention windows and transmission opportunity limits. Capitalizing on these, a consider-

able effort has been devoted to defining the most appropriate mapping functions between the set of end-to-end CoSs and those available in the considered network technologies: WMM, Ethernet/LAN 802.1p, Satellite, xDSL, UMTS, etc. Those mappings represent the basis for the *EuQoS QoS framework*, which is described in full detail in [34]. Two examples, related to Ethernet 802.1p and WiFi/WMM, are reported in Tables 2 and 3.

Depending on the capabilities of the underlying network technologies, the proposed solutions require an adequate Connection Admission Control (CAC) function to limit the QoS traffic, and a careful tuning of the available QoS mechanisms (schedulers, shapers, policers, etc.) in the network elements (IP routers, access points in WiFi, LAN/Ethernet switches, etc.). The latter has been performed through extensive simulation studies, as well as through field tests on the EuQoS prototype.

Each Autonomous System (AS) is free to offer transport services for a *subset* of the above end-to-end CoSs. In doing this, its administrator must be able to compute the QoS parameters related to the transit of traffic within the AS. In order to achieve a consistent forwarding treatment *across* AS boundaries, neighboring ASs negotiate per-CoS *peering Service Level Specifications (p-SLSs)*. The latter are bi-lateral agreements, specifying the amount and shape of traffic that the downstream AS (*provider*) is willing to accept from the upstream AS (*customer*), and the QoS that the provider AS is committed to provide to conformant traffic. The traffic submitted by the customer has to conform to a *Traffic Specification (TSPEC)* [10], while the QoS guarantees from the provider describe the target value of the QoS parameters to traverse the AS, from the ingress to the egress nodes. Traffic of a given CoS is not allowed to flow at an inter-AS boundary unless a related p-SLS exists. p-SLSs come with some kind of financial settlement between the customer and the provider, whose modeling is not in the scope of the EuQoS project.

Table 1

The set of EuQoS CoSs and the related DSCP.

EuQoS e2e CoS	DSCP name	DSCP value
Telephony	EF	101110
Signalling	CS5	101000
Real time interactive	CS4	100000
MMedia streaming	AF3x	011xx0*
High throughput data	AF1x	001xx0*
Standard	DF	000000
*xx={01,10,11}		

Table 2

Mapping between EuQoS e2e CoSs and Ethernet CoSs.

EuQoS e2e CoS	Ethernet CoS	802.1p priority
Signalling	Network management	7 (highest)
Telephony, RT interactive	Voice	6
	Video	5
	Controlled load	4
MM streaming, high throughput data	Excellent effort	3
	Undefined	2
	Background	1
	Best effort	0
Standard		

Table 3

Mapping between EuQoS e2e CoSs and WMM access classes.

EuQoS e2e CoS	WiFi CoS (WMM AC)	Target QoS parameters		
		Mean delay (ms)	Delay variation (ms)	Loss rate
Telephony, RT interactive	Real time (AC VO)	5	15	10^{-4}
MM streaming, high throughput data	Non-real-time (AC VI)	10	–	10^{-4}
Signalling	Signalling (SIG) (AC VI)	10	–	10^{-4}
Standard	Best Effort (AC BE)	–	–	–

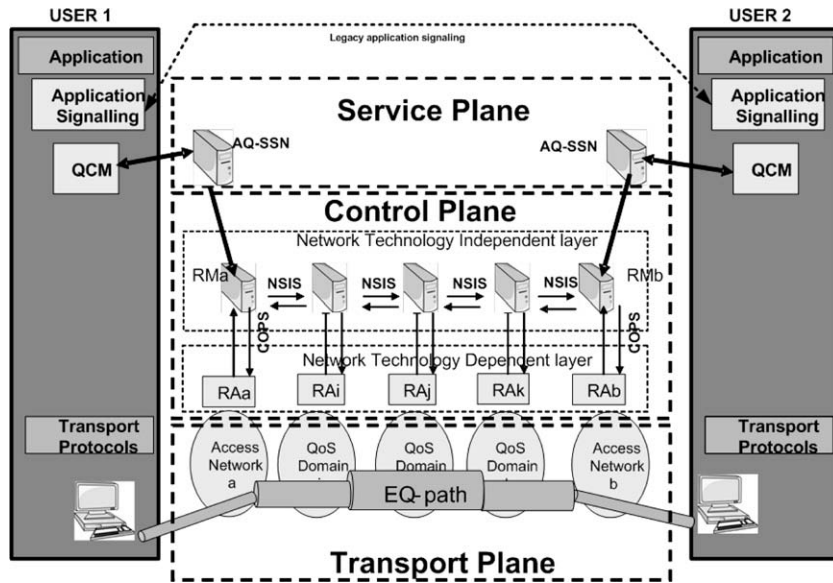


Fig. 1. High-level view of the EuQoS system.

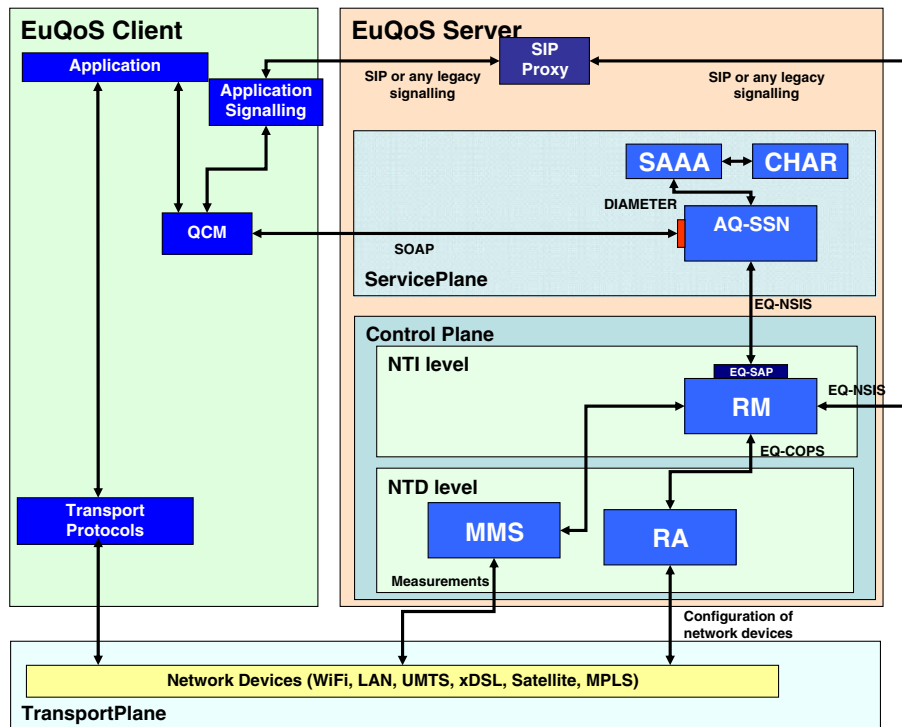


Fig. 2. Main components of the EuQoS system.

- The QoS-on-demand service is an interface exported by the Service Plane of the EuQoS system that allows the user to establish/release/modify a EuQoS session. It is designed as a multi-protocol interface and implemented through SOAP [25]. The current version of the QoS-on-demand can be used by the end user or by the administrator by means of a simple Web interface.
- The EuQoS Service Access Point (EQ-SAP) interface is exposed by the EuQoS Control Plane (described later on in Section 2.3), and it can be used by both the Service Plane and the operator-trusted terminals, in order to establish/release/modify QoS

requests per flow. It also supports request-response transactions and provides a reliable delivery of the messages. This interface is implemented using the Next Step In Signaling (NSIS) protocol suite [4].

Fig. 3 shows these two reference points and the agents that can use them.

As shown in Figs. 1 and 2, the user (client) side is supposed to run a Quality Control Module (QCM) that is in charge of interfacing to the EuQoS system. Depending on how the EuQoS services are invoked, we can classify the interactions as follows:

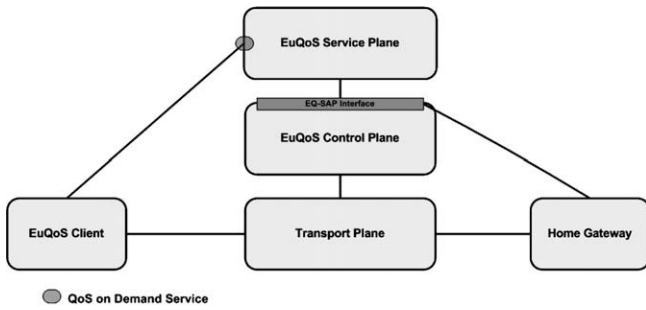


Fig. 3. Interactions between the EuQoS client and the EuQoS server.

- *EuQoS aware applications* have the QCM module embedded within. The QoS Request is sent to the *Application Quality of Service and Signaling Negotiation (AQ-SSN)* module when a specific application signaling event is detected. For instance, a SIP-based application will send a QoS Request when the SIP *Session Progress* message is received.
- For *non-EuQoS aware applications*, the QCM module is not part of the application. However, the latter can still be used as an independent process, running on the client hardware and directly managed by the user. Through the standalone QCM, the user can invoke the *QoS-on-demand* interface to request QoS guarantees for his own flows.
- *Operator-trusted terminals*, such as the Home Gateway, can directly use the EQ-SAP interface exposed by the EuQoS Control Plane to request QoS services. The Home Gateway embeds a service proxy Application Layer Gateway to manage the service in the user's Home Network, and in particular the firewall and NAT traversal. Such proxy can be used to trigger all the processes required to negotiate QoS with the underlying network on behalf of the applications, during the legacy service negotiation. For example, a Real Time Streaming Protocol proxy is used for VoD, while a SIP proxy or Back-to-Back User Agent (B2BUA) [39] is used for Voice, etc. More generally, the recent UPnP QoS v3 standard could also be used by terminals such as the *UPnP A/V Media Renderer* [38] to request QoS in the Home Network. In that case, the Home Gateway, acting as a QoS manager, can trigger the EuQoS QoS request to reserve bandwidth along the end-to-end path.

When a new QoS Request arrives at the AQ-SSN through the QoS-on-demand interface, the latter queries the *Security, Authentication, Authorization and Accounting (SAAA)* server using the *Diameter* protocol [15] to check whether the user is authenticated and authorized to request such a service. If the user request meets all the requisites, the AQ-SSN uses the EQ-SAP interface to request

the Control Plane to reserve the resources for the new call. Then, the AQ-SSN asks the SAAA to start the accounting associated to the ongoing call. When the session terminates, the SAAA sends the accounting records to the *Charging (CHAR)* module, in order to generate the bills associated to the session.

Note that, for some applications, a EuQoS service invocation can be asymmetric: for instance, a non-EuQoS-aware VoD client might require services from a EuQoS-aware VoD server, and the latter may invoke EuQoS services for a QoS-guaranteed video streaming.

When the QoS Request arrives to the EuQoS Control Plane, the latter takes care of routing the associated connection(s) along *End-to-end QoS Paths (EQ-Paths)*. In the next subsection, we shift our focus from the *user* side to the *network* side of EuQoS, and describe how an EQ-Path is built (*provisioning process*), used (*invocation process*) and monitored (*OAM process*).

2.3. The EuQoS Control Plane

The EuQoS Control Plane includes all the functions and protocols required to create, use and monitor end-to-end paths with associated resources, called EQ-Paths. We first describe how EQ-Paths are built, focusing on *QoS routing* and *resource provisioning*, and then describe how they are used and monitored.

2.3.1. Building the EQ-Path – QoS routing

In EuQoS, the EQ-Path is selected by the EQ-BGP interdomain routing protocol [2,3], an enhancement to the standard Border Gateway Protocol. EQ-BGP includes an optional path attribute, named *QoS Network Layer Reachability Information (QoS_NLRI)* that conveys information about the QoS parameters of a path, a configurable QoS-aware decision process for selecting the best end-to-end path also based on the QoS parameter values, and separate, per-CoS routing tables. EQ-BGP is configured by a *Traffic Engineering and Resource Optimization (TERO)* module, located in each AS. TERO configures the EQ-BGP decision process so as (i) to account for QoS parameters, and (ii) to select EQ-BGP updates that balance resource utilization. For instance, TERO configures EQ-BGP for updating the QoS_NLRI field of outgoing messages before they are advertised to another AS or inside an AS, so as to take into account the contribution of the domain to the QoS parameters related to a destination.

Fig. 4 shows how QoS_NLRIs are computed and advertised by EQ-BGP for a given CoS. Let Q_A , Q_B , and Q_C be the value of a QoS parameter (e.g., the delay) when traversing each domain, and $Q_{A \rightarrow B}$, $Q_{B \rightarrow A}$, $Q_{B \rightarrow C}$, and $Q_{C \rightarrow B}$ be the values related to the interdomain links. These values are configured on each EQ-BGP router by the TERO module in its AS. When C advertises a new destination to A through B, say NLRI_C, the QoS_NLRI attribute for that destination is progressively updated using the appropriate QoS composition function (a simple sum in this case). Thanks to this, A learns that

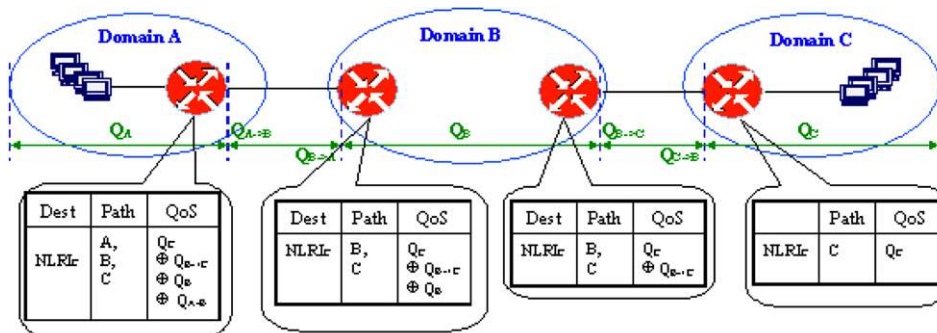


Fig. 4. EQ-BGP operation.

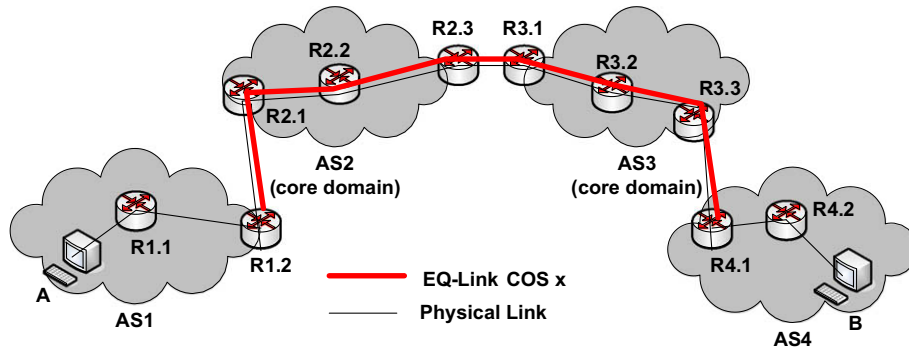


Fig. 5. An EQ-link between two remote networks.

the expected QoS on the path A–B–C to reach NLR_i is equal to $[Q_C \oplus Q_{B \rightarrow C} \oplus Q_B \oplus Q_{A \rightarrow B}]$, with \oplus representing the generic QoS composition function.

TERO routing decisions are taken either as a reaction to the changes in the network topology (e.g., negotiation of a new p-SLS with a neighboring AS) or periodically, for maintenance and optimization.

Being a minor modification of standard BGP, EQ-BGP can be easily implemented in routers. However, for those technologies which do not support EQ-BGP or for domains where EQ-BGP is not pertinent, the solution is to setup an EQ-BGP route reflector as a stand-alone server. In this case, the *multi-hop* BGP option must be used to link the peering entities.

2.3.2. Building the EQ-Path – open resource provisioning: the loose and the hard model

Thanks to the EQ-BGP protocol, each AS knows the QoS parameters that are associated to the EQ-Paths reaching a given destination for each CoS. As far as resource provisioning is concerned, two models are defined in EuQoS, namely the *loose model* and the *hard model*. The two options, which are described hereafter, offer a different tradeoff between manageability and scalability, and they have been designed to coexist in the same framework. As a feature common to both options, resources are provisioned on a per-CoS basis (i.e., they are not related to single connections), and their provisioning takes place at timescales which are in the order of hours and days.

2.3.2.1. Loose model. In the so-called *loose model*, resources are independently provisioned in every AS, and, although provisioned per CoS, they are not bound to a specific EQ-Path (which may in fact encompass several ASs). Each AS makes its own provisioning independently, configuring resources (e.g., policers, queue rates and buffers, etc.) in its routers according to the p-SLSs negotiated with its neighbors. In such an uncoordinated framework, a connection that requires to traverse a given EQ-Path has to make sure that enough resources exist on that EQ-Path to guarantee the required QoS. This is achieved through signaling and Call Admission Control (CAC) at the time the connection is setup. The *invocation process* actually reserves the necessary amount of resources for the new connection along an EQ-Path. In the loose model, therefore, the resources required for establishing a single user connection along an EQ-Path are dynamically reserved, composed and associated to that connection by a CAC function during the invocation process.

The main advantage of the loose model is that it requires minimum coupling among the ASs along an EQ-Path. In fact, it only requires peering agreements between neighboring ASs, without any end-to-end concept (and related management requirements). As such, it can be considered as the basic Internet-wide model, which makes the EuQoS solution potentially applicable to any technology (including all the heterogeneous access ones) and suitable for all policies implemented by a provider. The main disadvantage of the loose model is the amount of signaling involved in the call setup/teardown process, due to the dynamic binding of resources to the EQ-Path. In fact, as we will show later on in Section 2.3.3, each

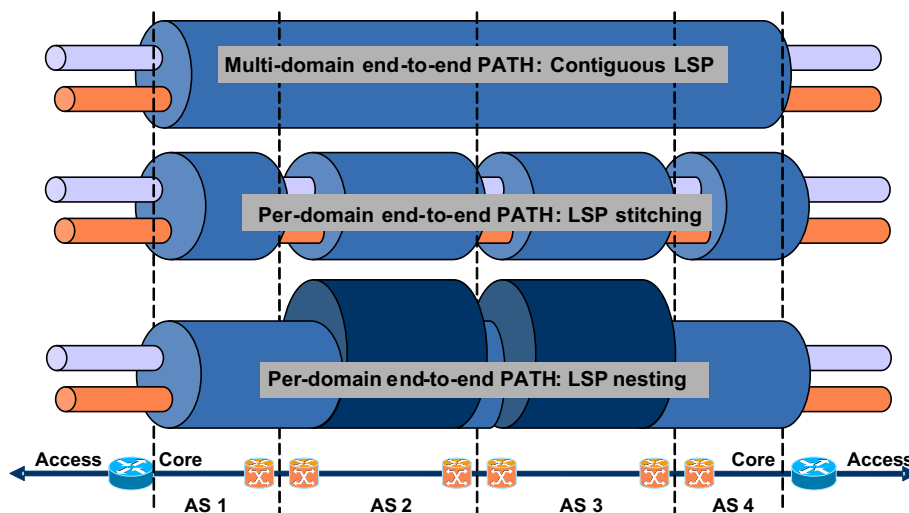


Fig. 6. Different styles of inter-domain LSPs.

AS in the EQ-Path has to handle signaling messages for each call setup/teardown. In a transit ASs, where the number of simultaneous ongoing calls is large, this may place a considerable processing burden on the servers.

2.3.2.2. Hard model. The *hard model* is based on the concept of EQ-link. An EQ-link virtualizes a link of known QoS characteristics between any two nodes in the network (not belonging to the same AS); as such, it is associated to a specific CoS – not to a session (i.e., it carries traffic aggregates) – and resources (bandwidth and buffer) are explicitly reserved for its *exclusive* use at provisioning timescales. In practice, an EQ-link can be setup as a DiffServ MPLS-TE [20] Label Switched Path (LSP) which may span over multiple ASs. Thus, it is semi-static, with resources associated to it, and it can be protected against failures. Based on this concept, an EQ-Path may be simply built, at provisioning time, by establishing a corresponding EQ-link across the Internet between two remote networks.

For instance, in the example of Fig. 5, AS1, AS2 and AS3 cooperate to establish an EQ-link between R1.2 (*head end*) and R4.1 (*tail end*). When a host in network A wants to open a connection to network B with CoS x , it is possible to use the EQ-link as a virtual link connecting R1.2 to R4.1. In this last case, AS2 and AS3 need not to know about the existence of the ongoing call between A and B, since resources are already provisioned for the EQ-link which transits through it as a whole: more specifically, CAC is only needed at the *head-end*, i.e., in AS1, instead of in each traversed AS. In general, an EQ-link may itself be the full end-to-end path between two access networks, or it may represent one (or more) *segment(s)* of the EQ-Path.

As far as inter-domain LSP setup is concerned, three different mechanisms have been identified by the IETF to setup MPLS tunnels crossing inter-domain boundaries ([21–23]): (i) the contiguous LSP, (ii) the LSP stitching, and (iii) the LSP nesting, all shown in Fig. 6. The latter is recommended due to its better scalability properties. In fact, each domain can setup a limited number of outer edge-to-edge LSPs (no more than a full mesh connecting all its border routers), and then stack an arbitrary number of EQ-links (themselves inner LSPs) within each outer LSP. This increases the scalability within transit domains, which may be traversed by a large number of EQ-links.

Since EQ-links are built with the frequency of the *provisioning process* cycles, their path computation is not subject to particularly stringent time constraints and can be made more refined. The computation of the EQ-link path is split into two separate phases: (i)

inter-AS path computation, i.e., optimal, or sub-optimal, inter-AS path routing for a given CoS and a given couple of ASs, and (ii) *full*, i.e., node by node, path computation. The inter-AS path is computed through the interaction of TERO sub-modules in neighboring domains through the *Inter-AS Path Computation Protocol (IA-PCP)* [24]. The computation takes into account *QoS requirements, resource availability, and administrative constraints* (i.e., p-SLS) that may limit the reachability of the destination with the CoS selected for the EQ-link. The inter-AS path computation follows a *source routing* approach, i.e., it is initiated by the head-end AS, which negotiates an inter-AS path to the destination AS by contacting neighboring ASs. This allows a head-end AS to constrain the path computation to specific criteria (e.g., max number of ASs to traverse, include/exclude certain ASs, etc.). The outcome of the inter-AS path computation, i.e., the list of ASs to traverse – called *Implicit Route Object (IRO)*, is made available at the head-end AS TERO module.

The *full* path computation is done by *Path Computation Elements (PCEs, [14,16])*. A PCE is a server which is capable of performing constrained path computation within a single area or domain for Traffic Engineering (TE) purposes. Furthermore, given an IRO, PCEs can also coordinate themselves so as to compute paths with a multi-domain scope. The IRO computed by the TERO module is thus fed to the PCE at the head-end AS. The PCEs in the IRO cooperate to compute the *full* end-to-end path, i.e., the *Explicit Route Object (ERO)* specifying all the nodes to be traversed. The ERO is finally used to setup the path using RSVP-TE. The whole process is shown in Fig. 7. The rationale behind using a two-phase path computation is to be able to take into account more information than the one that can be handled by the PCE alone, as for instance p-SLS constraints, administrative policies, traffic matrices, historical data on the level of resource utilization. Furthermore, a source-initiated inter-AS path computation can better exploit the multipath capabilities of a multidomain network, possibly exploring “good” routes which were not advertised through EQ-BGP (in EQ-BGP each AS only advertises *one* route per \langle destination, CoS \rangle , while more than one exist in general).

A good practical exploitation of the EQ-links would be to extend the reach of Virtual Private Networks (VPNs) beyond the domain borders. However, in EuQoS, they may represent either a whole EQ-Path, or a *segment* of the latter. In this last case, they act as *virtual links* connecting two remote endpoints. In this context, EQ-links can also be exploited to reduce the signaling message processing load in transit domains. This is explained at the end of the next subsection, after discussing the invocation process. Finally, we observe that the IP Sphere forum has proposed a framework and architecture for operators to negotiate and setup MPLS-TE tunnels at inter-domain boundaries. The EuQoS hard model is compatible with the IP Sphere specification, and could be used to establish end-to-end DiffServ-MPLS-TE tunnels with guaranteed QoS.

2.3.3. Using the EQ-Path – the invocation process

We now describe the actions that take place at call invocation. In doing this, we assume that the incoming connection traverses a number of domains in which resources are provisioned according to the *loose model*. At the end of the section, we discuss how using the hard model may simplify this process.

The QoS provisioning has to be coordinated among different ASs on the EQ-Path (represented by horizontal interactions in Fig. 1), and has to be translated into domain-specific, and often technology-specific actions (represented by vertical interactions in Fig. 1). As the primary goal of EuQoS is managing heterogeneity, the network functions in each AS are partitioned between a *Network Technology Independent (NTI)* and a *Network Technology Dependent (NTD)* layers. At the NTI level, *Resource Managers (RMs)* possess the AS-wide knowledge of the available and re-

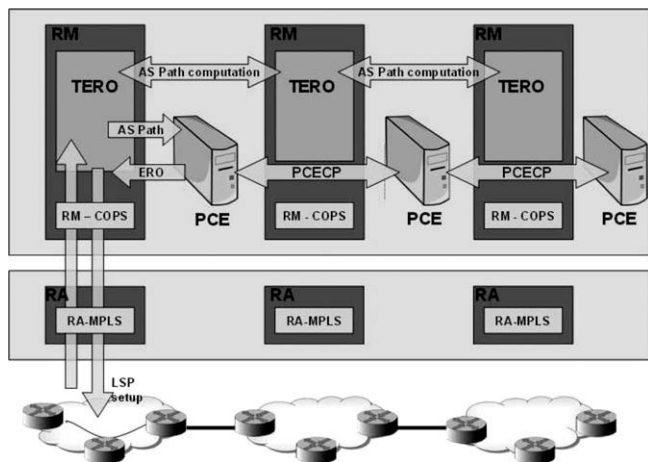


Fig. 7. EQ-link computation steps.

served network resources and of the routing topology. RMs are in charge of enforcing Call Admission Control (CAC) and they perform resource accounting (i.e., reservation and release) at flow setup/teardown. When the RM decides that the status of the resources needs to be changed, it communicates with one or more *Resource Allocators* (RAs) within its domain, lying at the border between the NTI and the NTD layers. The latter actually enforce the RM policies by configuring each specific underlying network device.

The RM receives a request for a QoS-guaranteed call to a given destination through its *EQ-Service Access Point* (EQ-SAP) interface, which is exposed to the Service Plane and to operator-trusted terminals. When such a request arrives, the RM does the following:

- (1) If the request has been originated within the local AS, the RM controls whether an EQ-Path of suitable QoS actually exists to the selected destination. This process, called *end-to-end CAC*, is in fact made possible because EQ-BGP conveys the QoS_NLRI, i.e., the QoS parameters on a route to a destination. The end-to-end CAC is thus simply a comparison between the QoS_NLRI value of the selected route to the destination, and the requested QoS parameters.
- (2) It selects the next downstream AS (*Network Selection*) to which the new call has to be forwarded. In doing this, it also identifies the outgoing inter-AS link and the intra-AS path. It then performs a *domain CAC* to determine whether sufficient resources are available on the links to admit the new call. This might imply interactions with local RAs. Furthermore, it also checks whether the incoming call request is compliant with the operator policies.
- (3) It contacts the RM in the next downstream AS to advance the call setup. The latter, in turn, performs steps (2) and (3) until the final destination is reached.

If all the RMs along the EQ-Path agree to let the call through, then the resources (which were pre-reserved by each RM during the above-described negotiation) are committed to the call, for its entire duration.

The main building block of the Network Technology Dependent (NTD) level is the Resource Allocator. The latter has a standard, technology-independent interface with the RM, and a technology-dependent interface with the various networking devices: for instance, EuQoS defines specific instances of RAs for WiFi Access Points, Ethernet switches [41], Cisco routers, etc. Thus, as new devices and technologies appear, the only thing that is needed to integrate them into the EuQoS architecture is to develop a corresponding RA interface. The main functionalities of the RA are:

- *Technology dependent QoS and priority mapping*: as explained in Section 2.1, *end-to-end CoSs* have to be mapped into technology-specific classes of service wherever appropriate. Such mapping is actually performed by the RA.
- *IP packet marking and rate control*: if a given technology can perform the above functions, the RA configures them according to the RM instructions.
- *Element resource control*: RA provides configuration and management of transport elements (e.g., routers) not only at aggregate level, but also per-flow if the access technology allows for it.
- *Technology dependent CAC*: The RA is able to perform technology-specific CAC algorithms, complementing those performed at the RM level. More specifically, the *domain CAC* takes place within the RM most of the times, without involving the RAs. If the CAC at the RM level fails, then the RM can query its RAs and have more refined CAC algorithms be performed at that level, in order to see whether the call can still be admitted. As a simple example, in a UMTS environment, the RM could check that the

requested bandwidth does not exceed 384 kbps and the UMTS RA could interface the GGSN to check whether there are enough resources in the network to establish a PDP context.

Fig. 8 shows the horizontal (i.e., inter-RM) and vertical (i.e., RM to RA) signaling exchange for the successful setup of a call traversing two domains, assuming that one RA exists in each domain for simplicity. The signaling protocols used for horizontal interaction is an extended version of the NSIS protocol called EQ-NSIS [5]. EQ-NSIS ensures that the signaling messages traverse the same AS-Path as the data (i.e., traverse the same ASs, and the same border routers within each AS), and each RMs receives the message from the ingress router (so that it is able to perform CAC before the signaling proceeds further). The signaling protocol used for vertical interaction is EQ-COPS, an extended version of the Common Open Policy Server protocol proposed by the IETF [6].

In the considered scenario, the first resource manager, RM1, receives a *Reserve Commit* request from the AQ-SSN server, describing the flow QoS requests. Next, it runs end-to-end CAC to see if a suitable path exists to the destination. Then it asks the CAC module in RA1 if enough resources exist to handle the new connection. If so, the requested resources are *Reserved* (i.e., booked) and RA1 sends an OK to RM1. RM1 then forwards the QoS request to its peer RM2 and sends a request to RA2 to actually allocate the reserved resources in the associated access network equipment. When all these actions have been accomplished, RA1 sends a confirmation to RM1. Finally, when RM1 receives the confirmation from both RA1 and RM2, it replies to AQ-SSN that the new call can be activated. The call handling scenario in RM2 and RA2 (i.e., in the egress domain) is the same as in RM1 and RA1 (ingress domain).

A call may obviously be routed along a path traversing more than two domains. In this case, transit domains are involved in the signaling, since resources have to be reserved in each domain. In that case, RMs in intermediate domains do not perform *end-to-end CAC*, whereas they only perform *domain CAC*. For bi-directional calls, the call handling process is performed simultaneously in the two directions.

Suppose now an EQ-link is provisioned across several domains, as in Fig. 5. As part of its setup process, resources have been reserved for its exclusive use in all the traversed ASs. Thus, if a call from AS1 to AS4 is routed through the EQ-link, domain CAC is *only* required at the head-end of the EQ-link, and not in the two transit domains AS2 and AS3. This allows EQ-links to be exploited for enhancing the scalability in large transit domains, whose RMs would otherwise need to handle a large amount of calls per second.

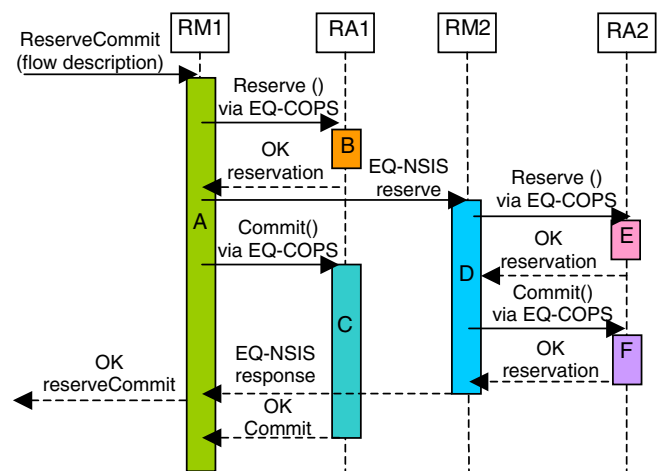


Fig. 8. Call scenario for two ASs, one direction.

2.3.4. Monitoring the EQ-Path – the OAM process

A correct resource provisioning and reservation scheme, such as the ones performed in the TERO, RM and RA modules, though essential, are not sufficient alone to guarantee a reliable QoS. In fact, faults may affect the QoS capabilities of a domain. For this reason, the *Operation, Administration and Maintenance (OAM)* process has been defined in EuQoS. The latter includes modules and functions to detect, report and correct faults and QoS disruptions, and is carried out at both the NTI and the NTD levels. The *Monitoring, Measurement and Fault Management (MMFM)* module stores and manages data related to the actual usage and QoS experienced on the domain links, working at the NTI level. The above information is conveyed by entities laying at the NTD level, which constitute the *Monitoring and Measuring System (MMS)* (see Fig. 9).

As an example, the *Topology Acquisition Tool (TAT)* within the MMS subsystem monitors inter-domain topology changes, interfacing to (EQ)-BGP routers or reflectors. When the TAT detects a change in the interdomain routing, it sends a notification to the MMFM, which in turn makes this information available to the rest of the modules involved in the invocation and provisioning pro-

cess. The data included in the MMFM database are also used by the TERO module for periodic resource optimization cycles. However, the MMS tools at the NTD level are also able to detect and communicate abnormal events (such as topology changes, abnormal resources usage, etc.) in real time. This allows the RM to react quickly to abnormal situations and minimize the duration of QoS disruptions.

While the MMS/MMFM system takes care of monitoring the status of the transport plane, the *Session Status Manager (SSM)*, located in the Service Plane of a source domain, monitors the status of ongoing sessions. The SSM receives periodic *keepalive* messages from the QCM of clients with ongoing sessions. If an application crashes, the SSM is thus able to detect its failure due to timer expiration. In that case, the SSM notifies the AQ-SSN to close the related reservations, stop the charging and free the associated resources.

3. Prototype implementation

This Section describes the EuQoS prototype implementation, also showing results from field trials.

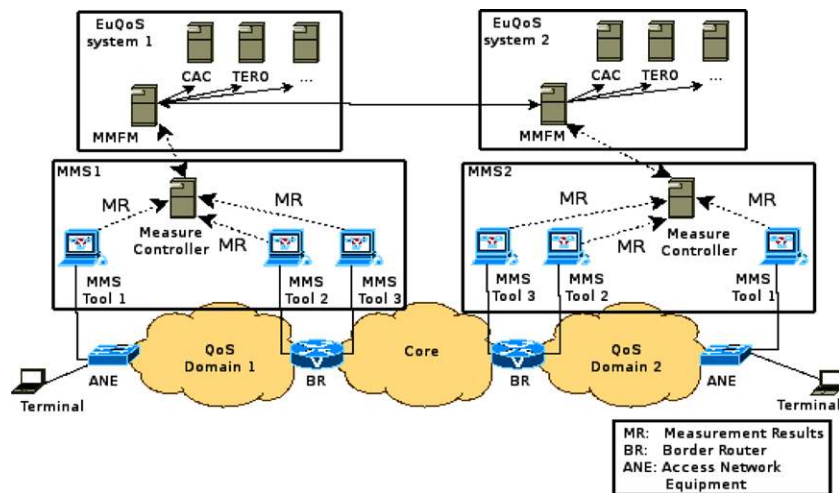


Fig. 9. The monitoring and measurement system.

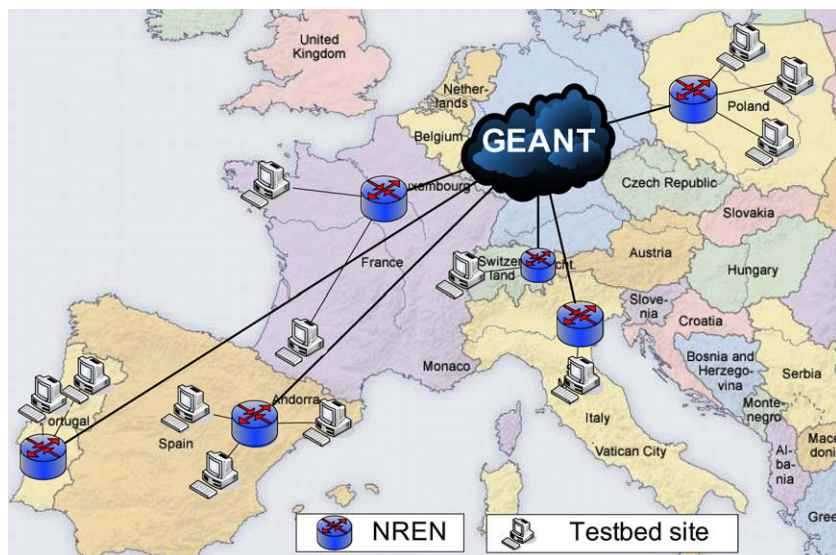


Fig. 10. Overview of the EuQoS testbed.

The EuQoS system has been actually implemented in a fully working prototype. Different prototype versions were released as project milestones, reflecting the progress in the specification of the architecture and system functionalities. Tests were carried out in order to provide feedback to the design process to optimize different processes and interactions.

The different prototypes were installed on the EuQoS pan-European testbed, shown in Fig. 10, consisting of twelve local testbeds connected to the respective National Research and Education Networks (NRENs) and interconnected through the GEANT European backbone networks. Each local testbed implements different access technologies (UMTS, WiFi, xDSL, Ethernet and Satellite). Some local testbed implemented multiple domains, using MPLS-TE in transit domains. In the prototype, the software modules described in Section 2 have been mainly implemented in Java, so as to allow an easy deployment over Linux-based platforms. The prototype was composed of more than twenty modules grouped in different deployment packages.

The final version of the software includes the complete stack of the service plane, interfaces to the end users (QoS-on-demand and EQ-SAP interfaces), the control plane with the provisioning and invocation processes functionalities (as explained in the previous section), roaming features, solutions to provide e2e CoSs in different technologies (xDSL, UMTS, Ethernet, WiFi, Satellite and IP/MPLS backbone). This final implementation of the EuQoS system demonstrated the correct synergy of the provisioning, invocation and OAM processes across different domains and through the different planes of the EuQoS system, as well as the effective cooperation of the different QoS mechanisms to provide end-to-end QoS across heterogeneous networks.

As part of the implementation several applications have been also integrated and tested with the EuQoS system. These applications have been selected in order to test the system response to different QoS requirements: real-time Applications such as VideoConference and on-line gaming, having strict delay and jitter requirements, and non-real-time applications such as a VoD, which instead require a small loss ratio. The prototype has also integrated existing Internet applications, such as the *Nexuiz* game [28], the *Teamspeak* VoIP application [29], the *Medigraf* telemedicine application [30] (this last one including synchronized audio, video, con-

trol and data streams), so as to provide proof of concept that applications can be easily adapted to (and benefit from) the EuQoS system.

We now report results from the field trials on the EuQoS testbed, showing the effectiveness of the devised architecture.

3.1. Field trials

The field trials were aimed at proving that the EuQoS system actually delivers end-to-end QoS in a multi-domain scenario with heterogeneous network technologies. The trials included the validation of the signaling system, the provisioning system and the QoS provided at the packet level in the various network technologies. The field trials have been complemented with an extensive simulation activity (see, for instance [31,32]), which also covered scenarios that could not be reproduced in the testbed due to scale or manageability reasons.

The following paragraphs describe the main experiments that have been performed.

3.1.1. Provisioning process validation

The trials related to the provisioning process focused on (i) validation of p-SLS negotiation; (ii) evaluation of EQ-BGP convergence; (iii) validation of the EQ-Link computation and setup. The trials of p-SLS provisioning proved that the interaction of the modules involved in the provisioning process works correctly: as soon as a new p-SLS for CoS x is installed on an interdomain link, the *customer AS* sets up a new queue for buffering CoS x 's traffic on the border router and enforces a rate which is computed coherently with the guarantees negotiated in the p-SLSs. On the other hand, the *provider AS* correctly configures policers at the entrance so as to protect itself from customer traffic surges, and allows EQ-BGP updates to flow across the interdomain link.

The trials related to EQ-BGP convergence were performed in different configurations of EuQoS testbeds, including a full mesh, a ring and a chain networks consisting of 4, 7 and 10 ASs. The tests show that EQ-BGP correctly establishes routing paths: the established paths offer the best possible QoS level. Furthermore, EQ-BGP correctly calculates the aggregated values of QoS parameters for each of e2e CoSs before advertising routes to the upstream do-

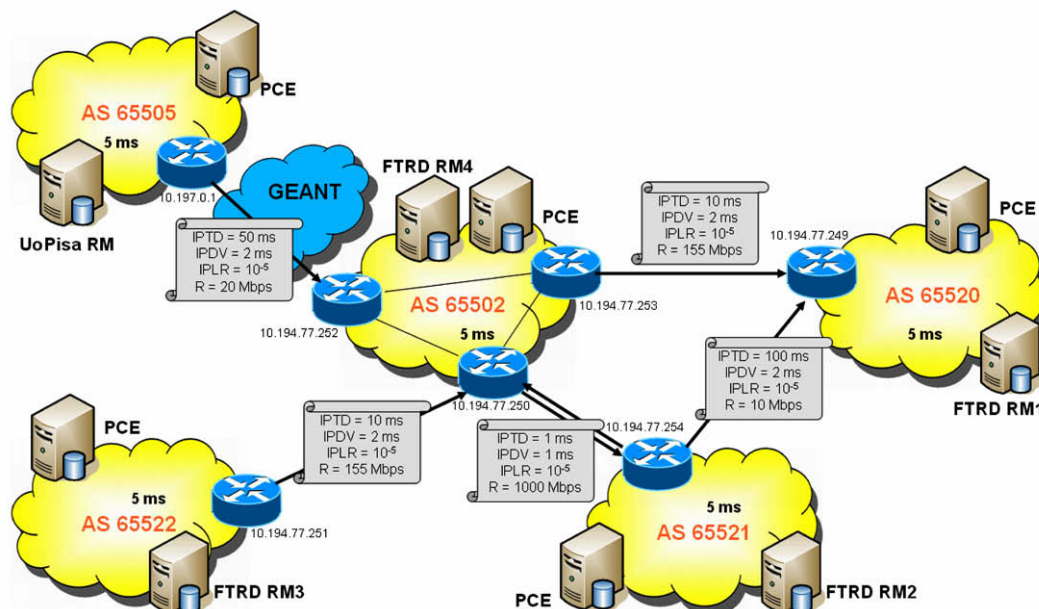


Fig. 11. Network scenario for measuring EQ-link setup times.

mains. Furthermore, EQ-BGP used as a route reflector correctly controls routing of CISCO and Juniper routers.

We have measured the time it takes to compute an EQ-link in the testbed, i.e., the time taken by the TERO and the PCE modules to compute the full path, plus the time required to notify the head-end RA about the MPLS-TE tunnel setup request. Fig. 11 shows the scenario used for the measurements, composed of five ASs. The figure also reports indications about the configuration of resources and QoS in the various nodes, as well as the p-SLSs installed at the inter-domain peering points (shown as contracts in the figure). The *Real Time* CoS was assumed as a test case.

Table 4 reports the computation times for some EQ-links. Such times are the order of magnitude of 1 s, which is an acceptable value, much smaller than the provisioning timescale (hours or days). The measures do not include the MPLS tunnel setup in the border routers, which is heavily equipment-dependent, ranging from a minimum of 100 ms to a maximum of 2 s. Other measurements on the France Telecom R&D testbed [40] show that the overall time to *setup* an EQ-link ranges from 1 to 10 s, which is again a reasonable time budget for the provisioning process.

3.1.2. Invocation process validation

Firstly, in order to evaluate the invocation process, a set of functional tests were carried out, validating the signaling chain defined in the previous section. These tests included the validation of several scenarios: (i) session setup and release with 1, 2 or more domains, (ii) failure scenarios where resources are not available in different domains, (iii) scenarios where the client crashes to check the correct behavior of the SSM in charge of managing the “keep alive” messages, (iv) roaming scenarios, etc. The complexity of these scenarios was increased as far as new functionalities were available in the different prototypes. This set of functional tests demonstrated that the designed and implemented invocation process works properly and meets all the requirements.

Additionally, in order to assess the EuQoS signaling in terms of delay and load, paying attention to the overall set up process in the involved protocols (SOAP, Diameter, NSIS and COPS) a set of additional tests were performed. These tests covered the successful session establishment and release in 2 or 3 domains. In order to address this goal, an ad-hoc tool was developed. This tool was deployed in the EuQoS servers and clients involved in the session setup in order to capture all the signaling messages. After this capture, the different messages were correlated in order to provide the latencies associated to the session setup: computation times required by the AQSSN, the RM, the SAAA and the RA and the transmission delays. These tests have been useful to detect implementation bottlenecks. All these measurements have shown that, as expected, the most important contributor to the setup latency is the time associated to the resource configuration (from 500 ms in WiFi to 2–3 s in UMTS); for this reason, the final invocation chain aims to parallelize these two processes. In most of the tested scenarios, the setup latency is around 1–2 s in 2 domains, (except in UMTS where just the time to establish a PDP context is around 2–3 s, due to current technology limitations), which was evaluated as acceptable.

Table 4
Measured EQ-link computation times.

EQ-link	AS-Path	Setup time (ms)
1	65505 65502	617
2	65522 65502 65521	647
3	65505 65502 65521	917
4	65505 65502 65520	1060
5	65505 65502 65521 65520	1088
6	65522 65502 65520	1162

Table 5 shows the average setup latencies in tests carried out, in scenarios composed of two domains with different access technologies. The measured values includes all the processes shown in Fig. 8, as well as the Service Plane processes (QoS request from the user to the EuQoS server and SAAA process time). For each test, the maximum partial per-domain configuration time is also highlighted, together with the domain where the latter was measured.

3.1.3. End-to-end QoS – objective assessment

Regarding the evaluation of the provided e2e QoS, both *objective* and *subjective* tests have been carried out: objective tests included measuring the QoS parameters, i.e., the transfer delay, jitter and loss ratio of the data flows, and the performance and response time of the QoS enforcement mechanisms available in the technologies where EuQoS was tested. The subjective assessment included conducting Mean Opinion Score surveys over a set of selected applications, such as VoD, VoIP, Nexuiz and Medigraf.

As an example of objective QoS assessment, we report results of trials on the EuQoS testbed related to the popular Nexuiz application [28], which was integrated within the EuQoS system. Nexuiz is a First Person Perspective (FPP) interactive game application. Its source code is written in C and it is available under an open-source license (GNU Public License). The Nexuiz application consists of two parts: client and server. The client is responsible for rendering the game state to the user by displaying a representation of the virtual world. The game state is coordinated by a single server, that handles connections from up to 32 users. The client-server communication runs on UDP, and it is characterized by a relatively low bit rate (up to 128 kbps in each direction). Being an FPP game, Nexuiz favors users who demonstrate higher responsiveness to the changes of the game state. In such a “reaction based” competition, near-optimal network conditions are of paramount importance to ensure actual playability of the game: in fact, unbalanced round-trip times and/or packet losses would disfavor some users, making the game unfair and hardly enjoyable. Furthermore, the message round-trip time should never exceed few tens milliseconds in any case, in order not to become comparable to human reaction times. One hundred milliseconds are normally considered the maximum tolerable round-trip time for this case. Taking into account the above requirements, it is foreseeable that such an application would reap the full benefits of absolute QoS guarantees.

The Nexuiz client behavior was modified so that the latter issues a EuQoS resource reservation request during the connection setup with the server. If the request fails, the connection is closed. In order to establish a EuQoS session, the application specifies the following information:

1. IP addresses and UDP ports of both the client and the server – they are needed to classify packets in the network as belonging to the appropriate CoS.
2. IP address (possibly known through DNS query) of the AQ-SSN server. The latter is in fact the interface point with the network signaling.
3. Amount of requested resources: peak bit rate was set to 128 kbps.

Table 5
Average latencies measured during the invocation process.

Scenario	Setup time (s)	Max configuration time (s)
Ethernet-xDSL	1.72	0.81 (LAN)
xDSL-WiFi	1.55	0.65 (xDSL)
WiFi-UMTS	3.81	2.58 (UMTS)

4. End-to-end CoS – the traffic generated by Nexuiz is submitted to the *RT Interactive CoS* [34], which is engineered to ensure the following target end-to-end QoS parameters: mean packet delay < 100 ms, jitter < 50 ms, packet loss ratio $< 10^{-3}$, thus being compliant with the above requirements.

The aim of the trial was to verify the QoS compliance of the *RT Interactive CoS*, which handles Nexuiz traffic, in the so-called *worst-case traffic conditions* (i.e., when the traffic load approaches the admission control limit). Fig. 12 shows the network topology for the trial in the EuQoS testbed. The application server is located in the LAAS domain (Toulouse, France) while the application client is connected to the WiFi (IEEE 802.11e) access point in the WUT domain (Warsaw, Poland). The inter-domain link is simulated through a symmetric Premium IP tunnel service in GEANT with a 10 Mbps guaranteed bit rate. Moreover, each domain includes EuQoS nodes AQ-SSN, RM and RA servers that configure and manage the network devices in order to support end-to-end QoS guarantees (not shown in the figure for better readability). The configuration of QoS mechanisms (traffic conditioners, packet schedulers, etc.) was aimed at providing separation of resources for CoSs (link capacity and buffer space) in both the inter-domain link and the WiFi access point. In this trial we considered one CoS with QoS guarantees (*RT Interactive*) and one Standard CoS (uncontrolled best effort traffic).

We passively collected the traffic directly on the terminals using a *packet capture tool* (*tcpdump* in the Linux server and *Wireshark* on the Windows client). Then, using a dedicated processing tool we compared the traces, so as to extract the matching packets and to calculate the appropriate metrics. The clocks of both the cli-

ent and server terminals were synchronized with the GPS clock (with less than 1 ms skew) through the Network Time Protocol (NTP). Thus, by extracting the timestamps of matching packets from client and server traces, we were able to compute a reliable estimate of the QoS parameters.

The end-to-end QoS parameters related to the Nexuiz traffic were measured in three scenarios:

1. *Scenario A* (unloaded best-effort network). The network carries only traffic generated by the Nexuiz application. This scenario serves as a baseline for comparison.
2. *Scenario B* (loaded best-effort network). The Nexuiz traffic is carried using the Standard CoS. The network handles both Nexuiz traffic (foreground traffic) and artificially generated background traffic. The scenario revealed two bottlenecks: (1) the inter-domain link, in the client-to-server direction, and (2) the WiFi access point in the WUT domain, in the server-to-client direction.
3. *Scenario C* (EuQoS network). The application reserves resources with the EuQoS system in *RT Interactive CoS*. The network handles foreground (application) traffic and artificial background traffic, both in *RT Interactive CoS* and Standard CoS. At both the above mentioned bottlenecks, the traffic load in the *RT Interactive CoS* is close to the maximum admissible load according to the admission control rules [42].

Table 6 shows the measured packet transfer characteristics. In Scenario B, both the delay and loss ratio values are unsatisfactory, due to the high load of background traffic. On the other hand, in Scenario C, the EuQoS mechanisms (resource reservation, admis-

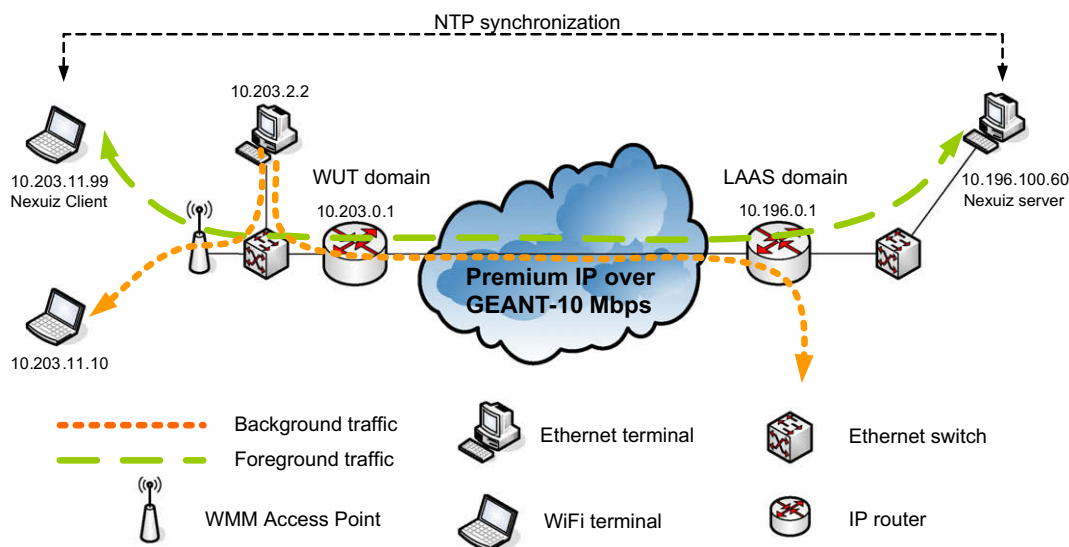


Fig. 12. Trial network topology.

Table 6
Trial results for Nexuiz.

Scenario	Direction	Mean delay (ms)	Delay variation jitter (ms)	Packet loss ratio	Packets sent
A	WUT → LAAS	28.1	6.9	No losses observed	27,868
	LAAS → WUT	28.7	8.9	No losses observed	30,025
B	WUT → LAAS	358.6	9275	5.4×10^{-2}	34,632
	LAAS → WUT	46.0	551.9	2.9×10^{-3}	36,284
C	WUT → LAAS	30.7	28.4	6.2×10^{-4}	26,004
	LAAS → WUT	29.5	17.3	No losses observed	28,509

sion control, traffic scheduling and buffering at the nodes) ensure a predictable and satisfactory performance, even in the presence of background traffic. Compared to Scenario A, where no background traffic is present, only the delay variation exhibits significantly different values. Nevertheless, all metrics conform to the target values defined for the *RT Interactive* end-to-end CoS.

3.1.4. System performance assessment

While the tests reported in the previous subsection were aimed at assessing the objective/subjective QoS capability, it is also relevant to assess the performance in terms of setup delay and blocking probability of EuQoS connections. These metrics are relevant not only to validate the scalability of the proposed approach, but also to provide a set of recommendations for the commercial implementation/deployment of the EuQoS system.

The above performance tests have only covered the EuQoS modules involved in the invocation process: Service Plane (AQ-SSN + SSM + SAAA) and Control Plane (RM and RA). They did not include the validation of the application signaling level, which is not strictly part of the EuQoS system. In these trials, the time spent in performing the specific NTD algorithms/configurations has been assumed equal to 700 ms, taking into account the outcome of the tests performed during the invocation process.

The test results (see Figs. 13 and 14) have shown that in an environment composed of Pentium III and IV hosts with no more than 1 Gb of RAM, considering that the 99-quantile of the setup latency cannot exceed 7 s for a single domain, it can be stated that the system is able to support call rate of 20 sessions/s; this working limit is due to a limitation in the memory consumed by the system. The system is able to support 4000 simultaneous connections.

Currently, a Mobile Switching Center (MSC) must support around 10 calls/s; thus, these tests demonstrate the viability of the system. Although these figures may not be enough for a large scale deployment, they are quite promising for a prototype. Indeed, a boost of a factor 10 with respect to the above figures can be achieved by just re-engineering the code and using higher-end servers. Furthermore, it has to be taken into account that a large, transit domain in the core network can benefit from using edge-to-edge MPLS-TE tunnels and hierarchical EQ-links, so as to reduce the amount of signaling that it needs to handle within its own EuQoS servers.

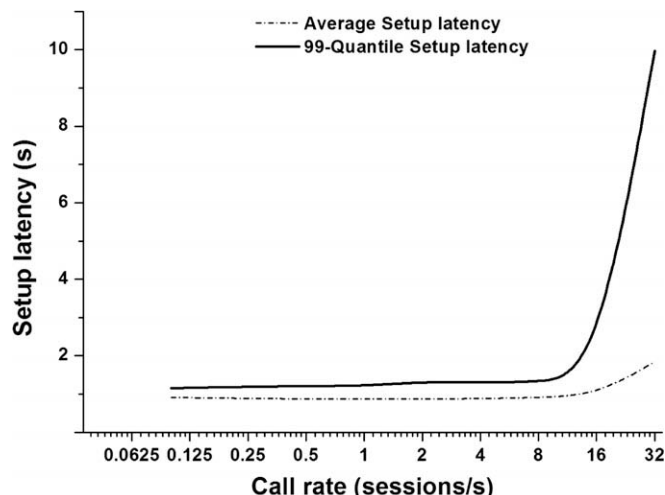


Fig. 13. Setup latency of a EuQoS call.

4. Discussion and open issues

In this section we enumerate the points of strength of the EuQoS architecture, discuss differences and compatibility with the IP Multimedia Subsystem (IMS) [26], developed within the framework of the Third Generation Partnership Project (3GPP), and address deployment issues for large-scale scenarios.

The following have been recognized as points of strength of the EuQoS architecture:

- (1) *Clear specification of the interfaces*: the interfaces between: (i) the EuQoS client and EuQoS server; (ii) the Service and Control planes, and (iii) the NTI and NTD levels are standardized and based on standard protocols whose performance has been thoroughly tested. This makes EuQoS easy to use from a client standpoint, amenable to legacy as well as new applications, and easily extendable to both other services and other network technologies.
- (2) *Net Neutrality*: the user is free to choose – through the QoS-on-demand interface – different levels of QoS for his/her application. In other words, applications and QoS are independent.
- (3) *Feasibility from a business perspective*: Internet Service Providers are not required to change their bi-lateral peering models in order to support EuQoS.
- (4) *Amenability to incremental deployment*: not all the Internet is required to switch to EuQoS simultaneously in order to reap the benefits of the EuQoS system. Specifically, if the backbone is reasonably overprovisioned (as it currently is), deployment in some access domains alone, coupled with either manual configuration of (IP or MPLS) tunnels between EuQoS domains, or routing overlays, will improve the service for the EuQoS traffic of the involved domains, without having significant impact on the rest of the Internet.

4.1. Comparison with IP Multimedia Subsystem (IMS)

We have already anticipated in Section 1 that the main difference between EuQoS and IMS lies in the fact that the latter requires SIP to interact with the main coordination point, i.e., the Proxy Call Session Control Function (P-CSCF). This is likely to become a major impairment in the development of IMS. In fact, on one hand, if SIP is discovered to be unsuitable for accommodating future applications, this would make the whole IMS control framework unusable to them. On the other hand, this does not allow existing applications using different signaling protocols to use the IMS framework. For instance, popular applications using their own protocols (e.g., Skype, MSN, P2P streaming, etc.) would require *application gateways* in order to be integrated with SIP-based IMS. Such gateways might end up being performance bottlenecks, and for sure would render the system complex, making it difficult to compose different services and hindering the traditional openness of the Internet. On the contrary, EuQoS does not impose such requirements on service negotiation, thus being amenable to incorporating legacy and non-SIP-based applications.

Having described the EuQoS architecture in great detail in Section 2, we are now able to draw a functional comparison between EuQoS and IMS, showing that the former is in fact compatible with the latter. The architecture of IMS is described in several documents in [26].

Some functions introduced by the IMS architecture are supported by the RM and RA modules. In particular, the *Access- and Core Resource Allocation and Control Functions* (A-RACF and C-RACF) within the *Resource Allocation and Control Subsystem* (RACS) are

supported by the RM and RA modules. However, in the EuQoS system the QoS demand is not carried through SIP, but through a dedicated SOAP message or through NSIS.

Furthermore, as specified in the Resource Allocation Control Function (RACF) standard [43], EuQoS divides QoS control and enforcement between two layers, namely network technology-dependent and -independent. The PD-FE (Policy Decision Functional Entity) functions are performed by the RM and the TRC-FE (Transport Resource Control Functional Entity) functions by the RA. The main difference lies in the protocols used by EuQoS. In fact, *Rt*, *Rw* and *Rc* interfaces are implemented through COPS, whereas *Rd* and *Ri* through NSIS. Charging and AAA, which are taken into consideration within the ITU-T NGN model [44], are also supported by EuQoS. Finally, the TERO and PCE modules, which implement the EuQoS *hard model*, are compatible with the recent ITU-T standards [45,46], which describe how the RACF interacts with MPLS-TE.

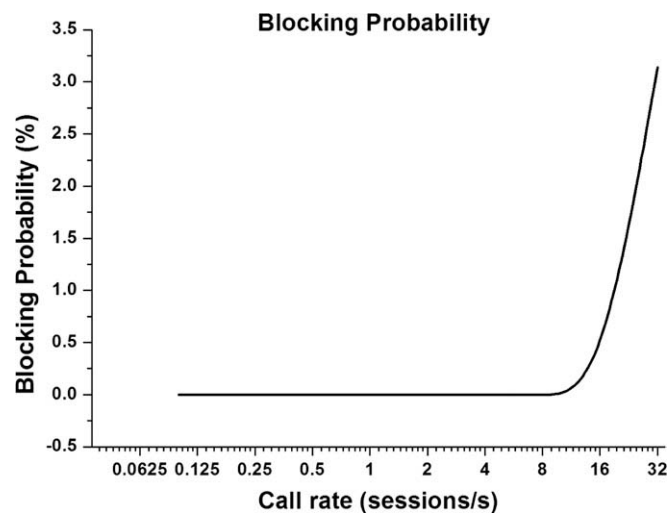


Fig. 14. Blocking probability of a EuQoS call.

Fig. 15 shows a diagram of the IMS functional blocks, on each one of which the corresponding EuQoS functions are mapped.

Hereafter, we discuss the issues in deploying the EuQoS system in a larger scale in the Internet, and we outline two likely deployment scenarios.

4.2. Deploying the EuQoS system on a large scale

Given the stated mission of providing end-to-end QoS in wide-area multi-domain heterogeneous networks, the question about the scalability of the EuQoS system is of major importance. Below, we will summarize the results and considerations related to the various aspects of the EuQoS system, that show promising in that respect.

Besides being the crucial element in the EuQoS architecture, the RM is also the crucial point for assessing scalability. With respect to that, distributed implementations for large domains have been investigated within the framework of the project. RAs, which are in charge of specific networks within a domain, can be replicated at will, so that they do not represent a scalability problem. Per-flow configurations are kept to a minimum within EuQoS. The CoSs represent aggregates of traffic in a DiffServ style, and per-flow actions (such as policing and shaping) are only taken in relatively small access networks (e.g., WiFi Access Points).

As far as the *invocation process* is concerned, simulation studies reported in [31,35], show that the main processing burden lies with the *application signaling*, which is handled *only* in the end (access) domains, and limits the call rate in those domains to few units per second, which is however sufficient for their scale. On the contrary, a *transit domain* should be able to handle up to 150 new calls/s and assure that for 95% of these calls the set-up latency is below 11 s, the latter being a requirement for user acceptability [31]. As far as the *provisioning process* is concerned, the scalability of EQ-BGP is similar to that of BGP-4, and this was proved by simulation and confirmed by experimental results [2,3].

These results show that the EuQoS system is a scalable proposal for assuring end-to-end QoS in the future Internet.

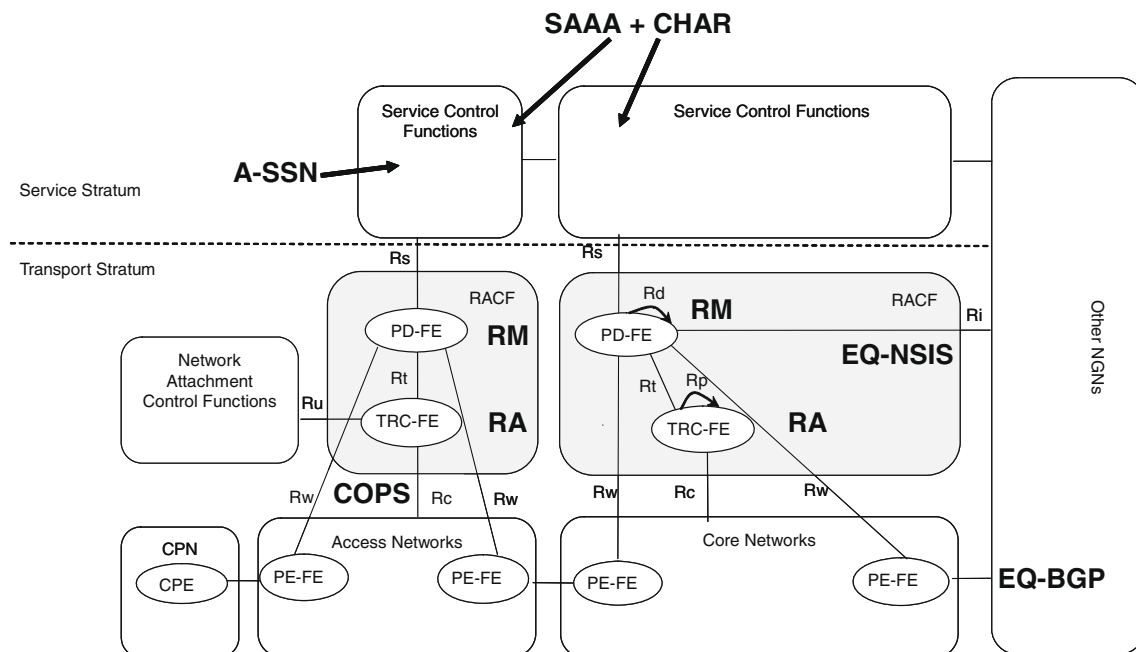


Fig. 15. Functional comparison between EuQoS and IMS.

4.2.1. Deployment case studies

Leveraging on the expertise of European telecommunication operators as project partners, several scenarios for the actual deployment of the EuQoS system for commercial and production purposes were envisaged. We report here two case studies, namely VoD Service deployment in an xDSL network and a fixed-mobile convergence scenario.

4.2.1.1. VoD Service deployment in an xDSL network. This first proposed scenario intends to provide a VoD Service in xDSL network operator. In this scenario, shown in Fig. 16, the usage of EuQoS

could be interesting to guarantee the delivered QoS of the Videos. The figure shows the deployment scenario.

In this network scenario, the nodes to be taken into account in this point are the CPE (Customer Premise Equipment), the DSLAM, the aggregation switches and IP Edge or BRAS. Using the EuQoS system, it would be possible: (i) to configure the Home Gateway using the defined interactions, (ii) to apply QoS policies and CAC algorithms associated to the DSLAM and (iii) configure the prioritization in the IP edge node.

Assuming a client population of 7 millions, a service penetration of 50% and one movie per user/ per month during peak hours, it is

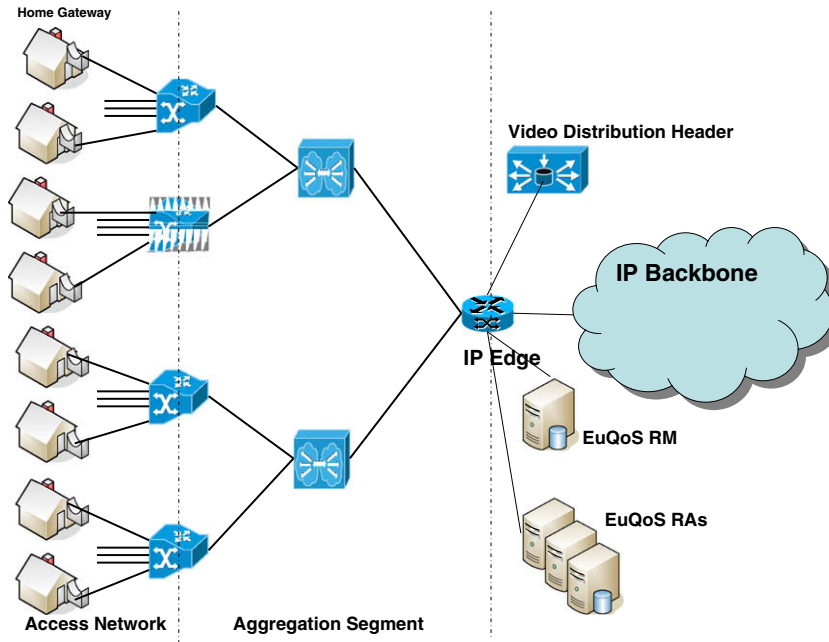


Fig. 16. VoD deployment scenario.

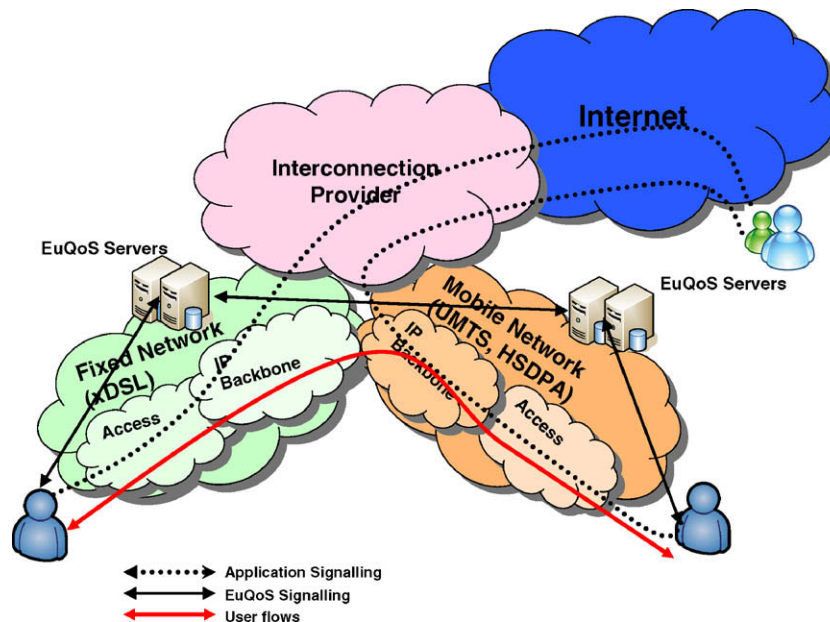


Fig. 17. EuQoS in fixed-mobile convergence scenario.

expected that the number of requests per second during the peak hour will be around 10 sessions/s. This requirement can be met by the current EuQoS system implementation, as shown in Section 3.1.

4.2.1.2. Fixed mobile convergence scenario. The main EuQoS strength is to coordinate the QoS mechanisms of different underlying network technologies. For a convergence operator (managing both a fixed and a mobile network, e.g., xDSL and UMTS), deploying EuQoS could enable QoS guarantees for connections crossing the two domains. A partial deployment of the EuQoS system in those two domains would have no impact on the interactions with other neighboring domains. As shown in Fig. 17, EuQoS users of this convergence operator could enjoy end-to-end QoS guaranteed connections in their networks. In this example, we assume that two users located in the different access networks would like to use a legacy videoconference application available in today's Internet. While the application signaling usually traverses the standard Internet, in order to, e.g., locate the other user, the data path will only traverse the operator's fixed and mobile networks, since the two ends of the communication are in the fixed and mobile domains of the integrated operator. This way, one user will ask its associated EuQoS server to reserve end-to-end resources for the data flows. Since, in this convergent operator network, both the fixed and mobile networks support EuQoS, the resources will actually be reserved.

5. Conclusions

The EuQoS system allows Internet Service Providers to offer a reliable differentiated QoS to the end user, meeting Net Neutrality requirements. Following the Internet approach, the main attribute of the architecture proposed in this paper is its openness: on one hand, its QoS framework and subdivision into a Network Technology Independent and Network Technology Dependent layers allow the architecture to accommodate the currently most popular access technologies, as well as to easily integrate new ones as they emerge; on the other hand, the EuQoS can be invoked by non-EuQoS aware applications through the QoS-on-demand service, without requiring any change in the application signaling. This paper has described the EuQoS system, focusing on its QoS model, architectural components and resource provisioning styles. The effectiveness of the proposed architecture is testified by the results obtained in a prototype implementation. On one hand, the latter confirm the expectations related to the actual QoS perceived by the users, even in the presence of heavy cross-traffic. On the other hand, they show that, although within the computational limitations of a prototype, the EuQoS Resource Manager architecture is able to support a suitable load for a real large-scale network environment.

Some issues still require to be investigated before making EuQoS a commercial product. More specifically, security issues, fault tolerance and compliance with NATs still have to be worked out.

Acknowledgements

The research work presented in this paper has been developed within the framework of the EuQoS Project, partially funded by the European Commission under the Information Society Technology (IST) priority within the Sixth Framework Program (Contract IST-004503).

We would like to thank all the EuQoS consortium partners for their contribution to the project. Specific thanks go to Luís Cordeiro of University of Coimbra, Florin Racaru of LAAS-CNRS, and Luca Bisti of University of Pisa, for their contribution to the implementation.

References

- [1] The IST-EuQoS consortium. Available from: <<http://www.euqos.eu/>>.
- [2] X. Masip-Bruin et al., The EuQoS system: a solution for QoS routing in heterogeneous networks, *IEEE Communications Magazine* 45 (2) (2007) 96–103.
- [3] A. Beben, EQ-BGP: An Efficient Inter-domain QoS Routing Protocol, in: Proc. 20th Int'l. Conf. Advanced Info. Networking and Apps., Vienna, Austria, April 2006.
- [4] Next Steps in Signaling (NSIS) IETF WG. Available from: <<http://www.ietf.org/html.charters/nsis-charter.html/>>.
- [5] L. Cordeiro, Hybrid on-Path Off-Path Approach for End-to End Signaling Across NSIS DOMAINS (HyPath), IETF draft-cordeiro-nsis-hypath-00, February 2006.
- [6] D. Durham, The COPS (Common Open Policy Service) Protocol, RFC 2748, January 2002.
- [7] RFC 4594, Configuration Guidelines for DiffServ Service Classes, August 2006.
- [8] IST-EuQoS deliverable D1.1.1, Business models and system design specification, 2005.
- [9] IST-EuQoS deliverable D1.1.3, Definition of Business, Communication and QoS models, 2005.
- [10] RFC 2216, Network Element Service Specification Template, September 1997.
- [11] RFC 3261, SIP: Session Initiation Protocol, June 2002.
- [12] RFC 4566, SDP: Session Description Protocol, July 2006.
- [13] ITU-T Recommendation Y, General Overview of NGN, 2001. Available from: <<http://www.itu.int/>>.
- [14] RFC 4655, A Path Computation Element (PCE)-Based Architecture, August 2006.
- [15] RFC 3588, Diameter Base Protocol, September 2003.
- [16] RFC 4657, Path Computation Element (PCE) Communication Protocol Generic Requirements, September 2003.
- [17] IST-TEQUILA project. Available from: <<http://www.ist-tequila.org/>>.
- [18] IST-MESCAL project. Available from: <<http://www.ist-mescal.org/>>.
- [19] QBONE project. Available from: <<http://qbone.internet2.edu/>>.
- [20] RFC 4124, Protocol Extensions for Support of Diffserv-aware MPLS Traffic Engineering, June 2005.
- [21] L. Fang et al., Interprovider IP-MPLS services: Requirements, Implementations and Challenges, *IEEE Communications Magazine* (2005), June.
- [22] RFC 4726, A Framework for Inter-Domain Multiprotocol Label Switching Traffic Engineering, November 2006.
- [23] RFC 4206, Label Switched Paths (LSP) Hierarchy with Generalized Multiprotocol Label Switching (GMPLS) Traffic Engineering (TE), October 2005.
- [24] O. Dugeon, E. Mingozi, G. Stea, L. Bisti, Provisioning QoS in Inter-domain Traffic Engineering, *Springer Annals of Telecommunications*, vol. 63, Springer, 2008.
- [25] W3C Recommendation, SOAP Version 1.2 Specification Assertions and Test Collection, second ed., April 2007.
- [26] IMS forum website. Available from: <<http://www.imsforum.org/>>.
- [27] IPShere website. Available from: <<http://www.ipshereforum.org/>>.
- [28] Nexuiz official website. Available from: <<http://alientrap.org/nexuiz/>>.
- [29] Teamspeak official website. Available from: <<http://www.goteamspeak.com/>>.
- [30] Medigraf website. Available from: <<http://www.medigraf.pt/>>.
- [31] H. Tarasiuk et al., Designing the simulative evaluation of an architecture for supporting QoS on a large scale, in: Proceedings of QoS2008, Marseille (FR), March 3, 2008.
- [32] C. Cicconetti et al., Simulation model for end-to-end QoS across heterogeneous networks, in: Proceedings of 3rd IPS MoMe 2005, Warsaw, 2005, 79–89.
- [33] Cisco white paper "Approaching the Zettabyte Era", June 2008. Available from: <<http://www.cisco.com/>>.
- [34] W. Burakowski, et al., Provision of end-to-end QoS in heterogeneous multi-domain networks, in: *Springer Annals of Telecommunications*, vol. 63, nos. 11–12, Springer, November–December 2008.
- [35] W. Burakowski, et al., Report on scalability evaluation of EuQoS system, EuQoS consortium, January 2008. Available from: <www.euqos.eu/>.
- [37] The GÉANT Network: <http://www.geant.net/>.
- [38] Media Renderer architecture. Available from: <<http://www.upnp.org/>>.
- [39] SIP Back-to-back user agent. Available from: <<http://www.b2bua.org/>>.
- [40] O. Dugeon et al., Prototype P#4 tests report", Deliverable 5.2.3, IST/FP6 EuQoS consortium, December 2007. Available from: <<http://www.euqos.eu/>>.
- [41] M. Silva Carmo, J. Sa Silva, E. Monteiro, EuQoS approach for resource allocation in Ethernet networks, *International Journal of Network Management* (2007).
- [42] ITU-T Recommendation Y.1541: Network Performance Objectives for IP-Based Services. Available from: <http://www.itu.int>, February 2003.
- [43] ITU-T Recommendation Y.2111: Resource and admission control functions in Next Generation Networks. Available from: <<http://www.itu.int/>>.
- [44] ITU-T Recommendation Y.2012 ERTA 1: Functional requirements and architecture of the NGN release 1. Available from: <<http://www.itu.int/>>.
- [45] ITU-T Recommendation Y.2174: Distributed Resource and Access Control Function (RACF) architecture for MPLS core networks. Available from: <<http://www.itu.int/>>.
- [46] ITU-T Recommendation Y.2175: Centralized Resource and Access Control Function (RACF) architecture for MPLS core networks. Available from: <<http://www.itu.int/>>.
- [48] IST-AGAVE project. Available from: <<http://www.ist-agave.org/>>.
- [49] IST-DAIDALOS project. Available from: <<http://www.ist-daidalos.org/>>.