4 Control Layer Architecture view

First of all, the QoS request is exchanged at the Service plane by means of EQ-SIP messages which carry the QoS request inside the EQ-SDP blocks. The EQ-SDP message is setup by the ASIG module which resides in the application or in a signalling proxy and process by the A-SSN module which reside in the proxy server (see figure below). A detail description of the service plane could be found in [10] and [11].

4.1 The Resource Manager

The RM contains a set of modules articulated around a common database, named RM-DB, which store all pertinent information from the various modules which composed the RM. This includes the different SLS manipulated by the RM as well as topology and policy information. Three QoS and 2 signalling modules composed the RM as follow:

- CAC modules
- The Traffic Engineering and Route Optimization (TERO) modules
- Monitoring, Measurement and Fault Management (MMFM) modules
- RM-SSN which manages the signalling between the RM
- RA-SSN which manages the signalling with the Resource Allocator

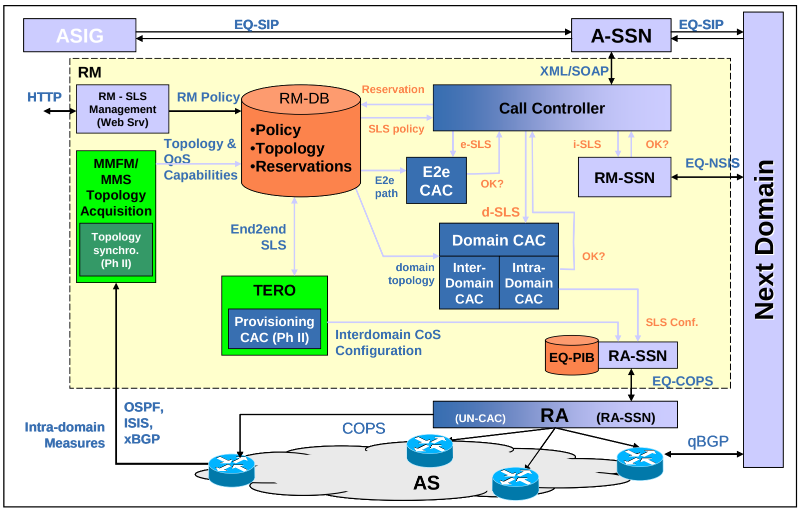

Fig. 5. RM architecture

4.1.1 Signalling blocks

NSIS [12] is used by the RM-SSN module. It was chosen as it is the natural upcoming IETF new signalization for IP. It is named EQ-NSIS because a new NSLP layer has been designed to carry EuQoS system message. This protocol suite will be used for resource reservation between the initiator and the receiver EuQoS systems, using the hop-by-hop paradigm of NSIS. NSIS will be the basis for signalling interactions between RM. Despite its relatively immature state in the standards bodies, the basic NSIS functionality is defined, which allows the development of a simplified version and adoption in the EuQoS architecture. But it is especially its path-coupled nature which decided to select NSIS because the RM must be in touch with the data path.

In EuQoS, EQ-NSIS takes care of all NSIS protocol interactions between peer RM-SSN modules; the main purpose of these interactions is the support of resource reservation and management along the data path across the various QoS domains. Decisions on resource reservation itself will be performed by the Call Controller module of the RM, which will interact with the various CAC functions for the purpose of its operation.

Because COPS was chosen as the protocol between the RA and device nodes for the configuration, this protocol was also chosen for the communication between the RA and the RM. Indeed, the RM pushed configuration into the device through the RA. These types of configurations are technology independent, but remain similar to those carried by COPS [13]. So, the RA-SSN acts as a PDP for the RA which implement a PEP function regarding the link with the RM. The RA also implements a PDP regarding the device nodes. An EQ-PIB database which contains the Policy send by the RM to the RA is also managed by the RA-SSN function.

4.1.2 Connection Admission Control (CAC)

The CAC function in the EuQoS system is probably the most difficult to design as it is here that the crunch point for resource distribution in the network occurs. The main goal of CAC is to check availability of resources. Different CAC are considered regarding the technology levels:

- Inter-Domain, Intra-Domain and End-to-End which are independent of the network technology. The resources availability checking is performed on the part of EQ-path which crosses a given AS domain. The RM-DB contains the resources associated to each path (EQ-path and domain path). CAC algorithm is a simple counting of QoS resources on the different paths.
- Underlying Network which is dependent on the network technology. The resources availability checking is performed by complex algorithms depending on the technology, from simple case, like LAN, xDSL, to complex one for UMTS, Satellite and WiFi.

In addition, and because CAC manipulates QoS parameters at different level under the form of SLS, a certain number of SLS have been defined:

- *SLS* : the QoS parameters give by the service plane to the RM
- *e-SLS* : the QoS parameters corresponding to the end to end part
- *i-SLS* : the QoS parameters of the inter-domain part

- *d-SLS* : the QoS parameters belongs to the domain own by the RM
- *r-SLS* : the QoS parameters give to the next RM

From this division, the CAC module was slice into four sub-modules:

- A coordinator named Call Controller which implements the CAC state machine and controls all CAC sub-modules.
- One devoted to the end to end, named e2e-CAC, in charge to select the EQ-path corresponding to the CoS contain in QoS parameters.
- One devoted to the domain, named domain CAC. This sub-module includes both the control of the intra-domain part and the inter-domain part by checking resources availability on the peer link.
- One devoted to the underlying network, named UN CAC. This sub-module is located in the Resource Allocator because it is dependent on the network technology like all sub-modules include in the RA.

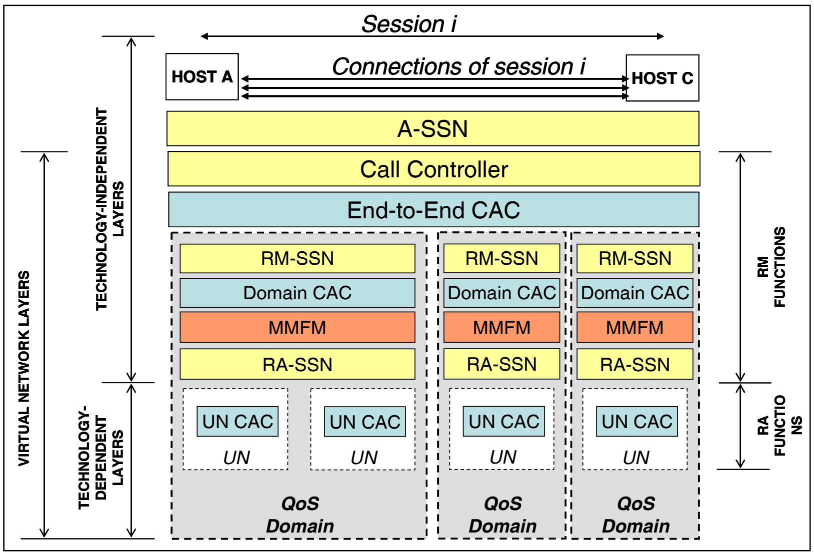

Fig. 6. EuQoS CAC Functions

End to end CAC is defined by checking if there is an EQ-path to the destination. Of course, this EQ-path is derived from the qBGP routing protocol. There might be a possibility that there is no EQ-path that meets the QoS requirements (the CoS part of the QoS). This can happen from different reasons, i.e. if all possible EQ-path have too much delay, jitter or losses regarding all parameters describing the QoS objectives. One example is when the application requires strong interactivity and when there is only a satellite link available to reach the remote site.

Inter-domain CAC. Even if an EQ-path exists, may be there is no enough resources available to ensure this QoS request along the EQ-path. Note that when an

EQ-path exists, it could be thought to be sufficient to also check that the requested bandwidth exists at that time in this EQ-path; nevertheless, this is not acceptable, as this capability implicitly means that there is a (end-to-end) tunnel from the sending domain to the receiving domain. Such EQ-path could be setup by means of end to end MPLS-TE. In the general case, we cannot assume that such a end-to-end tunnels with assigned bandwidth will exist, because keeping end-to-end tunnels does not solve the scalability problems (too many tunnels are needed and multiplexing gain is lost). As a consequence, in the general case, when QoS paths exist, their capabilities along the paths can be quite different: the existence of the requested resources along the path, in the consecutive inter-domain links has to be dynamically checked for the requests.

Intra-domain CAC is different for each domain. Intra-domain CAC can apply to QoS domains or AS constituted either by only an access network or by both an access network and a related core network. It is quite difficult to define a generic efficient solution for all technologies. The choice of the solution and of its implementation has to be left as a technology dependent matter. The decision has to be taken by the different technology providers, i.e. by the designers in charge of providing the QoS inside a given domain. Thus, the Intra-domain CAC has to trigger the RA, perform the network independent CAC, and integrate the results.

4.1.3 Global EuQoS CAC process

First of all, the Call Controller of the first RM receives an *a-SLS* from the A-SSN. This *a-SLS* carry all QoS parameters which include CoS, bandwidth, delay, jitter, loss… Then the Call Controller parses the *a-SLS* and extracts all information suitable to invoke the e2e-CAC by sending to it an *e-SLS*. The e2e-CAC checks if it exists a suitable EQ-path and sends it to the Call Controller. Again a parsing and extraction of the SLS is performed by the Call Controller to ask the domain CAC by means of a *d-SLS*. The domain CAC checks both intra-domain and inter-domain QoS resources before asking the different UN CAC located in the suitable RAs of its domain. Finally, the Call Controller receives an Ack (or a Nack) from its domain which embraces both dependent and independent network technology. At this moment, QoS is setup in the first domain. A final SLS parsing allow the Call Controller to produce the last QoS request in the form of a r-SLS and send it to the next RM through the RM-SSN signalling module. The SLS splitting follow the rules:

r-SLS(i+1) = r-SLS(i) – i-SLS(i)

where r-SLS(1) = a-SLS and i designs the i^{th} *RM.*

In subsequent RM, the Call Controller receives the *r-SLS* and start to interrogate only its domain CAC with the computed *i-SLS*. The e2e-CAC is no longer solicited by the subsequent RM since the EQ-path is selected once by the first RM. The process stops when *r-SLS = NULL* i.e. when we reach the destination domain.

At this step, the last Call Controller starts sending back the global acknowledge message. If a domain CAC failed, the process stops and a NACK message is immediately returned. The Call Controller removes the on-going QoS reservation or confirms it by storing it in the RM-DB.

So, the EuQoS system end to end CAC is based on a three level process as shown in figure below: a distributed end to end CAC performed by the e2e-CAC and the inter-domain checking of each domain CAC, an intra-domain performed by the domain CAC and a technology CAC performed by the each solicited UN CAC. The Call Controller, the e2e and domain CAC sub-modules use the RM-DB to store and manage the SLS and all suitable information suitable to check resources availability.

4.1.4 Traffic Engineering and Resource Optimization

Note that, in order to setup end to end path, we have to select an adequate set of QoS-related network reachability information (QoS NLRI) parameters values for defining this EQ-path. In particular, this QoS NLRI parameter value or values should not be too limited statically by some not powerful AS in the routes existing between the different domains, and should not be too large to loose bandwidth. Tuning these QoS NLRI parameters is a hard work and could give to EuQoS system a success or a failure.

So, inside the RM, and in the provisioning process, the TERO module is responsible to build the EQ-path in the best possible way. Its main objective is to control and optimize the routing process, so as to steer the traffic through the network in the most effective way, thus optimising the available resources when it builds the EQ-path. At the technology independent layer, traffic routes to be identified are between network domains (i.e., different Autonomous Systems), with the objective of optimising the inter-domain routing process based on QoS requirements. To perform this activity TERO interacts with qBGP router protocol in order to better configure it by giving them the most appropriate QoS NLRI.

It is assumed that this function does not directly control inter-domain or intra-domain resources, i.e. direct configuration of border and/or internal routers, but rather it provides a network administrator with the necessary information, hereafter called (with intentional generality) *policies*, to configure inter-domain traffic routes, which it carries out by means of the specific network technology dependent mechanism. Depending on the specific underlying network technology, the output of this function may also affect the intra–domain routing within the respective domain. In such a case, these policies can be provided also as input to intra-domain traffic engineering, whenever requested by a specific underlying network technology.

As a consequence, TERO works in background during the provisioning process (i.e. offline) with respect to the EuQoS system operation, its timeframe being that of *network engineering cycles* (i.e. hours, days or weeks) rather than the session lifetime, or round trip time.

4.1.5 Monitoring, Measurement and Fault Management (MMFM)

The goal of this function is to manage network measurement in order to monitor the network resource and discover the network topology to support CAC and TERO functions. It also performs fault management and QoS classes and SLA/SLS monitoring. To achieve these objectives, the MMFM architecture contains the following components:

- A test and a measurement subsystem to manage tests in a scheduled environment.
- A Web services in charge of starting the subsystems, in order to check the delivered end-to-end QoS.
- A set of probes in the network to collect test data.

As a first step in the function specification, the MMFM module:

- monitors the current status of the network in order to support the CAC decisions to accept/reject new session QoS requests,
- provides feedback to the TERO function,
- supports fault detection by monitoring the network elements and measuring their parameters in order to identify faults,
- notifies the service plane of any fault that can force a QoS level failure,
- monitors the level of QoS supported and delivered for determining if SLA/SLS requirements are met.

The network technology independent sub-layer of the Resource Managers defines what parameters must be measured. So, that the technology-specific Resource Managers are able to map these parameters to the technological specific ones, and vice versa.

5 Conclusion

End to End QoS is a difficult problem as the various QoS mechanisms for packet networks are being developed over ten years ago. The main reasons they have not been widely deployed is the absence of general architecture coordination, synchronisation with applications and business models.

Here we set out what is our first release of the architecture. We believe it is a most advanced result at this time but we do believe that we need to increase its maturity level to deal with the open issues, to incorporate feedback from the trials and in order to continue to simplify the system in order to minimise complexity and cost.

A first implementation – with Wifi and Ethernet as access network - has been released and show during the Communicating European Research (CER’2005) event in Brussels in November 2005. Next steps will refine and detail the architecture, in particular the “hard model”. Attention will be pay to scalability, IMS interoperability and performance. New implementations, including full MMFM support and first pan-European deployment, over 9 testbeds and 5 access network technology (Ethernet, Wifi, xDSL, UMTS, GMPLS) will be achieve during 2006 year.

6 Acknowledgements

The information drawn from this document is drawn from the EuQoS Consortium and individual contributors are too many to list. Particular thanks are due to people involved in CAC within the EuQoS consortium who edited much of the EuQoS material from which this paper comes from. A full version of this paper could be found in [11]

7 Bibliography

[1] FP6-IST/EuQoS project, (September 1, 2004 – August 31, 2007) ; http://www.euqos.org

[2] O. Dugeon & A. Diaconescu, *"From SLA to SLS up to QoS control: The CADENUS framework"*, WTC 2002, September 2002.

[3] S. Salsano,*"SIP Extensions for QoS support"*, draft-veltri-sip-qsip-01.txt.

[4] O. Dugeon and all, *"D6.2: Service Provisioning in Premium IP: Recommendations to Telecom Operators and ISPs"*, CADENUS, EU IST Project IST-1999-11017.

[5] *"eTOM Solution Suite"*; http://www.tmforum.org

[6] ITU-T Recommendation Y.1541, *"Network performance objectives for IP-based services"*, ITU, May 2002.

[7] M. Boucadair, "*QoS-Enhanced Border Gateway Protocol*", draft-boucadair-qos-bgp-spec-01.txt.

[8] FP5-IST/Tequila project (2001-2003); http://www.ist-tequila.org

[9] FP5-IST/Mescal project (2003-2005); http://www.mescal.org

[10] EuQoS Consortium, Deliverable D3.1.1, *"Extended QoS API and Middleware layer for phase 1 application use-cases"*, (not yet published), August, 2005.

[11] EuQoS Consortium, Deliverable D1.1.3, *"Business models and system design specification"*, (not yet published), August 2005.

[12] R. Hancock and all, "*Next Steps in Signalling (NSIS): Framework*", RFC4080, IETF, June 2005.

[13] Durham, Ed., J. Boyle, R. Cohen, S. Herzog, R. Rajan, A. Sastry, *"The COPS (Common Open Policy Service) Protocol"*, RFC 2748, IETF, January 2000.