End to End Quality of Service over Heterogeneous Networks (EuQoS)

O. Dugeon (FT/DR&D), D. Morris (RedZinc Ltd), E. Monteiro (UoC),
W. Burakowski (WUT), M. Diaz (LAAS-CNRS)
Email: Olivier.Dugeon@francetelecom.com

Abstract: In this paper, the EuQoS Consortium offers its first architecture release in order to provide an initial view on how end to end QoS is provided over multiple and heterogeneous networks. The EuQoS end to end Architecture has two views; a network deployment view across a number of autonomous systems (AS) domains and a software view within an AS. It was funded on a division of the end to end QoS paradigm along a vertical axis - Service, Control and Transport plane – and an horizontal axis – network division between the various technology especially Core and Access networks. The solution is based on the concept of end to end path build, used and managed by three processes: Provisioning, Invocation and Operating Maintenance (OAM).

Introduction

End to End quality of service support for multiple applications is required for the next major growth spurt in the telecommunications industry. With the increasing shift to the Internet Protocol for many networks and the desire by the telecommunications service providers to offer new value to their customers the need exists to finally coordinate the delivery of end to end quality of service so that providers may offer new services to support their customer's applications. The EuQoS [1] Consortium[1] offers its first architecture release in order to provide an initial view on *how this application-network-software* coordination might happen

[1] EuQoS is a consortium of organizations whose main objective is to research, develop, integrate, test, validate and demonstrate end-to-end QoS technologies to support advanced QoS aware applications – voice, video-conferencing, video-streaming, tele-engineering, educational and medical applications – over multiple, heterogeneous scientific, industrial and national research network infrastructure for real life users.

We provide an overview of the EuQoS architecture in Section 2 below. In Section 3 we describe the network view of the architecture by means of end to end path concept. And in Section 4, we detail the architecture of the various modules which compose the control plane of the EuQoS system.

EuQoS Architecture Overview

The EuQoS end-to-end architecture has two views; a network deployment view across a number of autonomous systems (AS) domains and a software view within an AS. The basic network deployment view is shown in figure below.

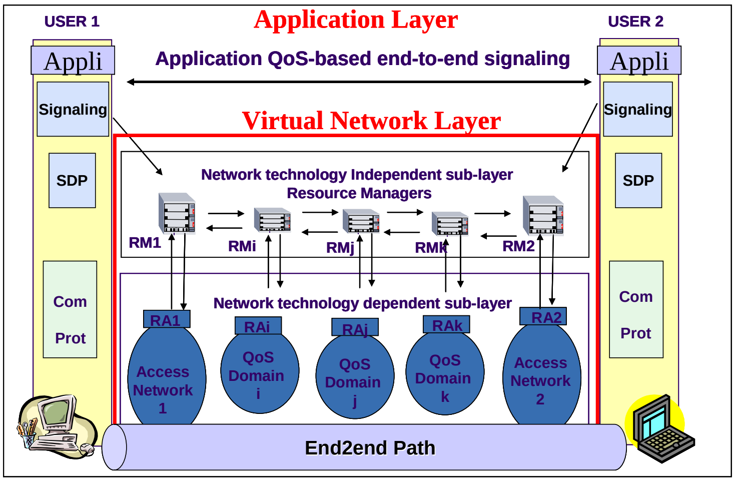

Fig. 1. EuQoS End to End Network Architecture

The EuQoS architecture approach to the scale challenge is the standard divide and conquers approach of separating the size of problem by splitting it into smaller parts. From the horizontal view, i.e. the different planes, this implies a clear separation between the Service (application layer) and Control Plane from the Transfer Plane which could be found in [2]. Again, the Control plane is sliced between a technology independent layer control by a Resource Manager (RM) and a technology dependent layer control by a Resource Allocator (RA). From the vertical view i.e. the different network partitions, this divide and conquer implies a clear separation between the various access technology and the different core network involved in the end-to-end connection.

The other ideas which govern our solution come from the synchronisation between service and control. Indeed, the control plane is solicited when an application needs some network resources. The aim of EuQoS architecture is not to provide end-to-end QoS for all applications, which is a significant challenge for large scale. But, it is rather to provide QoS for only applications which need them and only when they are needed. For this feature, the EuQoS system is based on session. First of all, application setups a session which triggers the corresponding network QoS setup. This has the advantage to perfectly synchronize the QoS requirement / setup and the usage of the QoS by the application. One other hard problem solved by the session concept is the *"graceful start release"* which allows synchronizing the end of the application and the release of network resources. Even if the application or the server crash, as soon as the session stops, or crashes, the QoS is released. In addition to the session setup, an application must be registered in the EuQoS AAA system. This allows the control of the application and the possibility to provide information to bill the end user. For this purpose, the EuQoS system used an enhanced version of SIP [3] which allows the QoS negotiation within the session establishment named EQ-SIP. The QoS request is simply transport in an extension of the SDP field, named EQ-SDP, of the SIP protocol. A dedicated SIP proxy handles this EQ-SDP and sends them to the Control.

In the same order, and after divided the problem by level and network partitioning, the EuQoS system divides and conquers the time scale. The architecture is based on three processes which are aligned with [4] and [5]:

- The provisioning process, whose function is to provision resources amongst the different AS, runs in background. It is the first process which starts the EuQoS system.
- The invocation process, whose role is mostly to perform admission control (CAC) of new connection, runs only when an application requests some QoS. It could be execute only when the provisioning process as finished its work.
- The Operating and Maintenance (OAM) process, whose function is to measure and monitor the EuQoS system, runs also in background. It provides the fault management sub-system. It interacts with the provisioning process in order to adjust or repair resources and with the invocation process in case of failure to notify the application.

Each process runs on a level (service and network) and for a given network partition or for the whole network (end-to-end).

End-to-end QoS path – EQ-path

Definition

The purpose of the EuQoS System is to build, use and monitor end-to-end QoS paths, named EQ-paths, with underlying QoS guarantees across multiple AS. The purpose of these EQ-paths is to support quality guarantees to applications on an end-to-end basis. Each EQ-path corresponds to a given set of QoS parameters and, in particular, of Classes of Service (CoS). If the number of CoS in a EuQoS system is under the control of the operators, it seems that a small number of different CoS will be sufficient for a wide range of application. For the EuQoS system, we have decided to based our CoS selection on the ITU-T Y.1541 recommendation [6].

EQ-paths have several properties which serve both the QoS guarantee and the scalability requirement. Indeed, building dedicated path on a per flow basis (a.k.a Intserv) is not scalable. At the opposite, just setup big trunk (a.k.a DiffServ) to carry a large amount of flow doesn’t guarantee any individual or strict QoS for a given flow. The EQ-path notion tries to combine advantage of DiffServ and IntServ while excluding their disadvantages. The EQ-paths follow the requirement of the EuQoS system. They are built, used and monitored by the control plane in the transfer plane, in order to address QoS needed by the service plane. In the same way, they are built to handle the Autonomous Systems (AS) paths with the clear separation between a wide variety of technology and network. The EuQoS system is responsible for choosing and joining each piece of network/technology path to form the EQ-path. Finally, the three different processes act as follows:

- The provisioning process is in charge to build the EQ-paths across network partition at both independent and dependent network level,
- The invocation process use the EQ-paths by selecting the most appropriate one and perform CAC to protect EQ-paths from overflows,
- The OAM process protects EQ-paths from failure and inter-act with provisioning and invocation processes to repair EQ-paths if needed.

Building the EQ-path (Provisioning)

Two main options could be considered in building an EQ-path: The multi-domain option and the per domain option. For the multi-domain option, an EQ-path is built from a source Access Network to a destination Access Network. For the per-domain option, the EQ-path is the result of the sequence of BR-to-BR path (where BR is a Border Router). The strengths and weakness of both are summarised below. The EuQoS architecture is based on the per domain option because it gives more freedom.

Multi domain	**Strength**: No need of Border Router (BR) configuration via a resource allocator, just the Access/Core router, in source and destination Access network, need configuration. Per flow CAC is done only in the access since the end to end path is well-known. **Weakness**: Need cooperation between all operators to build the EQ-path. Scalability is an issue since you must setup a full or partial full mesh of EQ-path. This is hard to maintain since it's difficult to provide a full backup path.
Per domain	**Strength**: Scalability is given since you can merge and aggregate traffic from one path to another. Flexibility is given too because each operator can setup its BR-to-BR paths independently, and it is easy to maintain by providing backup BR-to-BR path. **Weakness**: per flow CAC need to be performed by each domain Resource Manager (RM) to verify the traffic entering in each BR-to-BR path. The BR needs to be configured in each AS to connect the BR-to-BR path to form an EQ-path

Table 1. Per domain versus Multi-domain EQ-path

In addition to the per-domain path, EuQoS system enables aggregation and merging from one AS to another. This also allows a better scalability while setup fine paths and tune resources in the Access Network. The EQ-path could be view as a sum and concatenation of individual path setup in the different network partition. This avoids setup a partial or full mesh of EQ-path across all AS.

Since EQ-paths are end to end path, they cross several Autonomous System (AS). Because the Internet used BGP to connect Autonomous Systems, it is natural for EuQoS system to use it. The border routers creates a complex structure when building an AS path. Each AS may have multiple connections, some of which may be redundant connections to the same AS. The path determined by BGP-4 may not be the optimum path from a QoS or business point of view. Furthermore BGP-4 does not propagate QoS information, such as available bandwidth, AS delay or cumulative delay, packet loss, jitter. Such information is needed by the EuQoS system to determine the best AS path which could provide the requested QoS. The qBGP [7] scheme proposed by the IST/project Tequila [7] and developed by the IST/project Mescal [8] offers advantages as it propagates QoS information.

The principle of BGP is to convey information suitable to compute the AS path by means of NLRI (Network Layer Reachability Information). qBGP simply uses this NLRI field to convey information related to QoS From this new NLRI information, the AS path computation algorithm could determine an AS path based on QoS. In fact, qBGP just guarantee the continuity of a given Per Hop Behaviour (PHB) along an AS path.

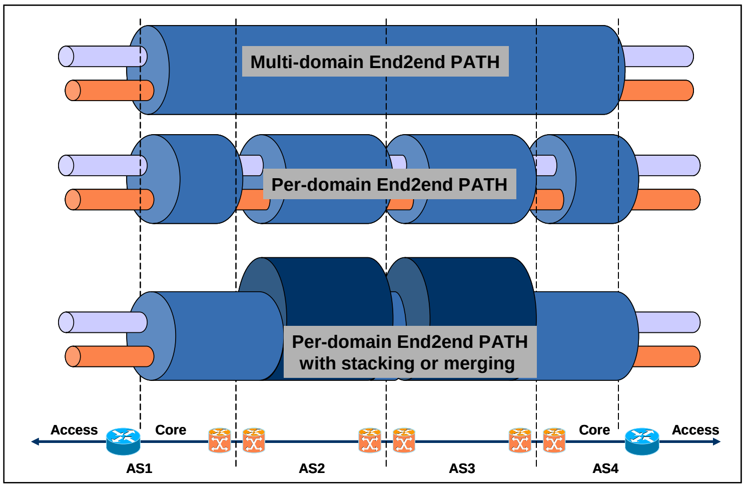
Multi-domain End2end PATH
Per-domain End2end PATH
Per-domain End2end PATH with stacking or merging
Access
Core
Core
Access
AS1
AS2
AS3
AS4

Fig. 2. Multi-domain versus per-domain option of EQ-path

If some of CoS satisfies a simple continuity of PHB, some others need strict QoS guarantees. These two models are well-know as the "*loose*" and "*hard*" model. The EuQoS architecture can be deployed in a loose model where the data path is determined by routing protocol on a hop by hop basis or in a hard model where the data path is established, a priori, by MPLS-TE in complement to q-BGP.

The key ingredients in building the data path are: qBGP, a Resource Manager (RM) which computes a set of QoS NRLI and qBGP enabled router.

- For the loose model the RM computes the QoS NLRI and configures the border router in order to convey this NLRI by the qBGP protocol. The RM and BR router, after exchanging their QoS NLRI capabilities, compute the best AS path and update their routing table.
- For the hard model the RM computes the QoS NLRI capabilities and exchanges them via qBGP. When each RM received all QoS NLRI, it computes the best AS path and enforces the route by means of MPLS-TE tunnel via the RA.

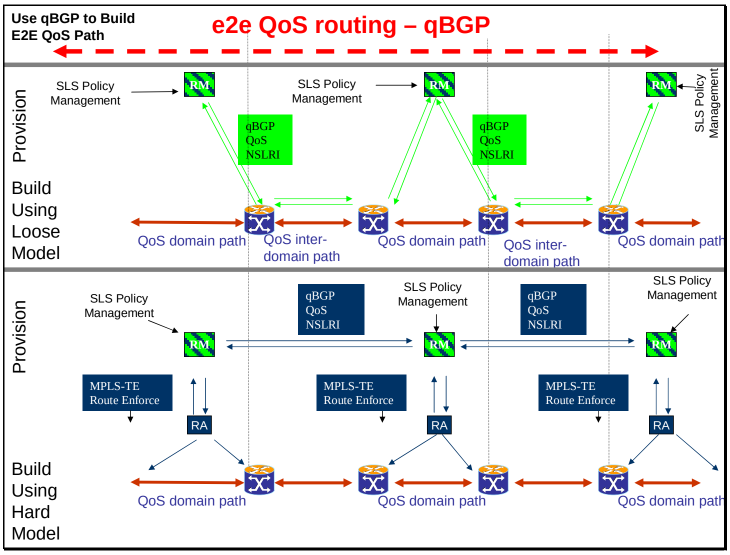

Fig. 3. Provisioning an EQ-path with Loose or Hard Model

Using the EQ-path (Invocation)

Once the EQ-paths have been built for each Class of Service, using either the hard or loose model, the invocation process can occur. A QoS request, which includes a CoS, is sent from the service plane to the first RM under the form of an SLS (Service Level Specification) during the SIP session establishment. The first RM corresponds to the one, in the chain, which is asked by the service plane and not by another RM. This first RM, which is responsible of the end to end QoS commitment, checks if it exists a suitable EQ-path regarding the requested CoS. This action in only done by the first RM since the EQ-path is selected once at the beginning of the path. At this point, the EuQoS system, in fact the RM, knows if there is a suitable EQ-path capable of deliver the requested QoS in term of CoS. The remaining part of the invocation process consist of verifying, that there is sufficient QoS resources in all AS, claim by the SLS, along this EQ-path. At this moment, each RM performs the same actions:

1. Perform resources checking for its own part of the EQ-path
2. Forward the remaining part of QoS request and the selected EQ-path to the next RM.

The next RM is automatically retrieves from the EQ-path information. Indeed, each RM manages its own database which, at least, stores all information around the EQ-path. This includes the next RM. So, hop by hop, each RM checks the resources availability along the given EQ-path. If the QoS could be meeting, QoS enforcement information is sent by the RM to the device nodes it controls (only those which need to be configured) through the Resource Allocator.

As describe in chapter 2, the EuQoS system is split between an independent and dependent network technology layer. The RM, which represents the independent network technology layer, performs its CAC on database information related to the route, bandwidth availability … Even if this information are linked to the reality, they are not the real state of the network. There are estimated, measured and/or computed resources, which for certain case is sufficient. So, at this step, the RA could also perform a CAC based on network technology dependent which represents the reality. Not only network resources are taking into account, but also constraints to enforce the QoS. When RA finishes its CAC and QoS enforcement, it sends an ACK or NACK message back to the RM. Then, the QoS request continues its trip to the next RM.

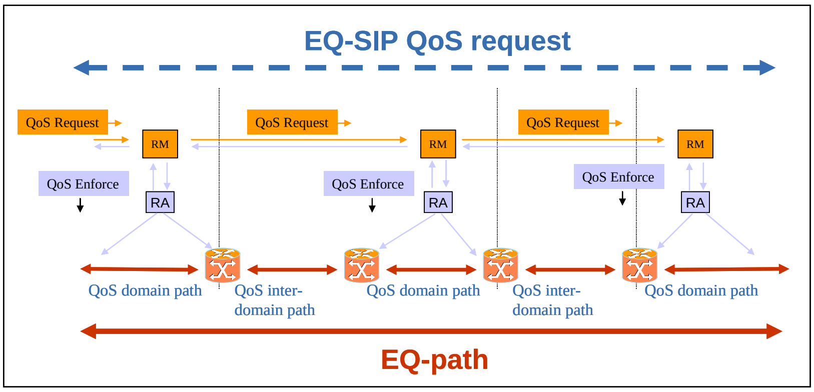

Fig. 4. Using an End to End QoS Path

When the last RM on the EQ-path receives the QoS request, it checks the resources and start sending a final acknowledge message to the previous RM. Again, hop by hop, each RM confirms the reservation for the previous one up to the first one. At this point, the QoS request has been transformed into a QoS reservation, stored in each RM database, and enforces in the needed device through the Resource Allocator. If one RM on the EQ-path failed to reserve resource, the reservation process stop and a NACK message is sent back. This has for effect to remove the pre-reservation that each RM made previously. Finally, the first RM aggregates all the answers and gives a final acknowledgement to the service plane which requests the SLS.

Monitoring the EQ-path

In order to guarantee the QoS commitment, the EuQoS system performs two actions: the first is the admission control given above, and the second is the monitoring of EQ-path. This is held by the OAM process. Monitoring is done by means of measurement and fault management. The measurement sub-system allows the EuQoS system to verify that EQ-paths are not overflow and used like expected i.e. the measured bandwidth corresponds, more or less, to the sum of reserved bandwidth. Fault management sub-system allows verifying the EQ-path continuity and takes care of device node and link failure. These two sub-systems interact with the invocation process, in fact the CAC, to adjust the admission control threshold, and the provisioning process in order to re-compute EQ-path in case of node or link failure. This path protection could also be improved by setting up some backup path when EQ-path is build with the hard model.

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].

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

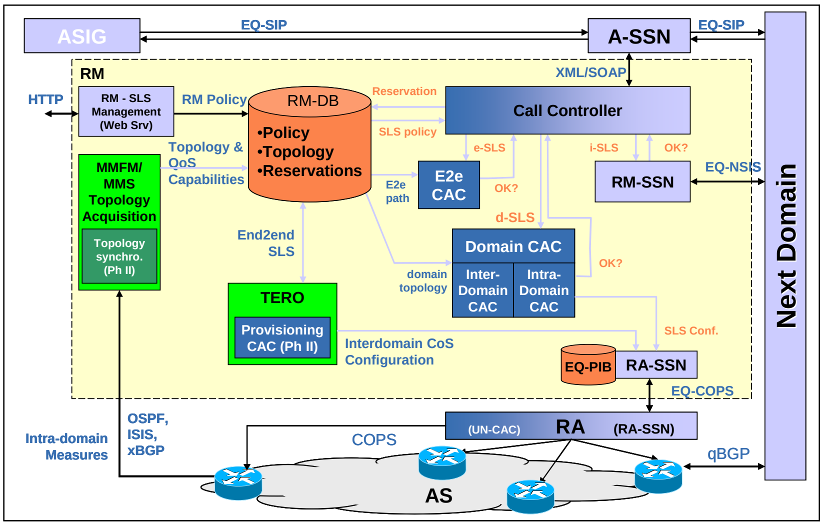

Fig. 5. RM architecture

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.

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.

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.

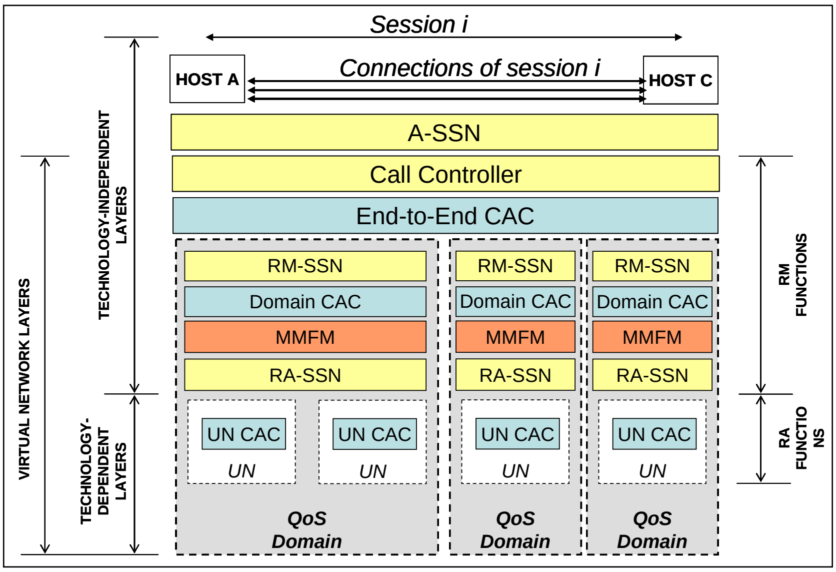

Fig. 6. EuQoS CAC Functions

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.

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:

$$r\text{-}SLS(i+1) = r\text{-}SLS(i) - i\text{-}SLS(i)$$

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.

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.

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,

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.

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]

Bibliography

[1] FP6-IST/EuQoS project 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] http://www.ist-tequila.org

[9] http://www.mescal.org

[10] EuQoS Consortium, Deliverable D3.1.1, *"Extended QoS API and Middleware layer for phase 1 application use-cases"*, August, 2005.

[11] EuQoS Consortium, Deliverable D1.1.3, *"Business models and system design specification"*, 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.