Network Independent Quality of Service

The role of Authentication, Authorization, and Accouting in a roaming environment

Jesús Miguel Guitérrez-Barquín

KTH Information and Communication Technology

Master of Science Thesis
Stockholm, Sweden 2006

COS/CCS 2006-7

Network Independent Quality of Service: The role of Authentication, Authorization, and Accounting in a roaming environment

Jesús Miguel Guitérrez-Barquín

Master of Science Thesis performed at
Department of Microelectronics and Information Technology (IMIT)
Royal Institute of Technology (KTH)
Wireless@KTH

Stockholm, Sweden
30th March, 2006

Advisor and Examiner: Professor Gerald Q. Maguire Jr.

Autor:	Jesús Miguel Gutiérrez-Barquín
Titulo:	QoS independiente de la red de acceso: comportamiento de AAA en entornos roaming.
Tutor:	Professor Gerald Q. Maguire Jr.
Institución:	KTH (http://www.kth.se)
Lugar de lectura:	Sala de conferencias Grimeton del centro Wireless@KTH, Isafjordsgatan 30B, Kista (Estocolmo, Suecia)
Fecha:	24 March 2006 at 11:00
Oponente:	Xiaoying Wang

Resumen del proyecto

Con la continua evolución de las aplicaciones basadas en el protocolo IP en todas las redes, y el deseo de los proveedores de servicios de telecomunicaciones de ofrecer un valor añadido a sus clientes, cohabita la necesidad de coordinar la entrega de calidad de servicio (QoS) extremo a extremo. De esta forma los proveedores pueden aumentar la oferta de servicios mediante nuevas aplicaciones.

El principal objetivo del proyecto EuQoS es investigar, desarrollar, integrar y probar una tecnología independiente de la red de acceso que garantice QoS extremo a extremo. El sistema esta pensado inicialmente para dar soporte a las aplicaciones: VoIP, VoD, video conferencia, y a una aplicación médica llamada MEDIGRAF, sobre múltiples y heterogéneas redes de acceso. Los parámetros que EuQoS tiene en cuenta para la reserva de la calidad de servicio son el ancho de banda, el retardo, la variación del retardo (*jitter*), y las pérdidas permitidas.

Un requisito fundamental para el modelo de QoS es que debe añadir la mínima complejidad posible al existe funcionamiento del sistema y debe ser compatible con el legado de aplicaciones y equipo. Esto se solucionará mediante el uso de señalización a nivel de Proxy.

Este proyecto analiza los posibles escenarios de *roaming* y cómo se debería afrontar la Autenticación, Autorización, y *Accounting* (AAA) en estas condiciones de itinerancia.

En los capítulos iniciales hacen una descripción general del sistema EuQoS, para tener una visión global del proceso de reserva de recursos. Es necesario conocer la estructura completa para lograr una integración mayor y con el menor coste posible.

El proyecto EuQoS propone y desarrolla un nuevo mecanismo de QoS que se construye sobre un estado del arte que incorpora los siguientes mecanismos: Monitorización y Medición, Control de Admisión, Gestión de Fallos,

Señalización y Negociación de Servicio, Seguridad y AAA, *Charging*, Ingeniería de Tráfico y Optimización de Recursos.

Con el fin de conocer lo que anteriormente otros habían hecho en este campo, antes de escribir una sola línea de este proyecto, llevé a cabo una extensa búsqueda de documentación. Parte de la información utilizada en este documento ha sido extraída de las entregas públicas del proyecto EuQoS hechas a la Comisión Europea. Además de la bibliografía mostrada en las referencias, Ericsson tiene sus propios informes técnicos e implementaciones de protocolos como el protocolo de iniciación de sesión (SIP) y DIAMETER, que se han consultado en varias ocasiones y han contribuido a lo largo de la investigación.

Existe una enorme similitud entre la arquitectura del sistema y la de IMS. De esta manera, algunos de los conceptos aplicados a la hora de desarrollar una solución para el caso de *roaming* para EuQoS se basan en los flujos de señalización utilizados en IMS y en los anteproyectos de nuevos RFCs.

Este proyecto consta de los siguientes capítulos:

- Capítulo 1: proporciona una introducción general a la tesis.

- Capítulo 2: es una visión general de los procesos de autenticación, autorización y *accounting*. Los temas más relevantes a cerca de AAA. Además, al final del capitulo hay una introducción al *charging* y se describen los CDRs.

- Capítulo 3: explica el protocolo DIAMETER y lo compara con RADIUS. Se evalúan en detalle las ventajas y desventajas que presenta el uno frente al otro.

- Capítulo 4: describe varias cuestiones sobre el sistema de tarificación, así como sus modelos *online* y *offline*.

- Capítulo 5: da una introducción al protocolo SIP. La terminología utilizada en SIP y sus métodos se describen en este capítulo. SIP se utiliza, junto con DIAMITER, en la comunicación entre el servidor AAA y el Proxy. A su vez, SIP también se ha elegido para llevar a cabo la reserva de recursos, pero se necesitan las precisas modificaciones para cumplir tal objetivo.

- Capítulo 6: explica el protocolo de descripción de sesión (SDP).

- Capítulo 7: describe la arquitectura de EuQoS. Se hace una primera aproximación al sistema que incluye el mapeado del esquema AAA con la arquitectura EuQoS. También en este capítulo, se describe el nuevo protocolo EQ-SIP.

- Capítulo 8: se representan los distintos casos de uso. Esta sección comienza con una introducción sobre *roaming*, en donde se plantean los distintos escenarios y finaliza con unos apuntes sobre *accounting* y su enfoque dentro de EuQoS.

- Los capítulos 9, 10 y 11 desarrollan los casos de uso expuestos en el capítulo anterior en detalle: un escenario en el que los usuarios se encuentran en su red de acceso por defecto (no hay *roaming*), otro en el que ambos están haciendo *roaming* y un último escenario en el que se comenta una situación en la que la red visitada no pertenece a EuQoS.

- El capitulo 12 analiza el diseño, muestra las conclusiones y el trabajo futuro que podría llevarse a cabo para mejorar y continuar el trabajo aquí mostrado.

- Finalmente, se incluye un apéndice en el que aparece la definición de los términos fundamentales utilizados a lo largo del proyecto, así como los acrónimos.

Conclusiones

Esta tesis describe varias propuestas viables para solucionar los diversos escenarios de *roaming* y propone soluciones para acometer la autenticación, autorización y *accounting*, así como la forma de tarifar los servicios (*charging*). Se ha elegido DIAMETER como el protocolo AAA ya que ofrece mejores prestaciones que RADIUS para el caso estudiado.

En EuQoS no se define un modelo de tarifas fijo, por lo tanto, esta tesis analiza varios modelos que se podrían emplear y explica por qué se debería implementar un registro tanto *online* como *offline*.

Se utiliza el protocolo SIP para realizar la reserva de recursos. Sin embargo, SIP no es capaz por si solo de realizar la reserva. Por este motivo, se propone un nuevo protocolo EQ-SIP, que introduce nuevos campos en la cabecera del paquete SIP y en la cabecera SDP. Estos campos posibilitan la negociación de los parámetros de calidad de servicio. La pila de SIP se mantiene y el nuevo protocolo EQ-SIP es completamente transparente a nodos que no pertenezcan a la red EuQoS.

Hay que tener en cuenta que cualquier proveedor de servicios operativo tiene su propio servidor AAA y por lo tanto será reacio a utilizar una nueva estructura. Esto implica que la solución propuesta tiene que integrase con la arquitectura existente de modo eficiente y con un coste moderado para facilitar la reutilización del sistema AAA de cada operador.

Todas las comunicaciones realizadas dentro del mismo dominio se llevan a cabo mediante mensajes SIP, es decir, no es necesaria ninguna comunicación externa entre sistemas AAA. Esto ayuda a simplificar la infraestructura del operador. Sin embargo, la inmediata desventaja es el incremento de la complejidad de la propia señalización SIP.

Una de las mayores desventajas de la arquitectura de EuQoS es que actualmente no se efectúa ninguna reserva de recursos en las redes intermedias. En esta primera fase de desarrollo se asume una red sobre-dimensionada, la red GEANT, pero esta aproximación se aleja de la realidad cuando se considera una red de capacidad menor. Medidas sobre la red GEANT muestran un pequeño incremento del retardo debido al tránsito por dicha red que está en torno a los 2 ms. Este valor puede ser significativo en ciertas aplicaciones que requieran grandes prestaciones.

Los datos necesarios para la autenticación y la autorización están almacenados en una base de datos del sistema AAA de la red del operador en la que el usuario hizo su suscripción (*Home*). Por lo tanto, toda petición de autenticación o autorización ha de llegar a esta red. Esto introduce un nuevo retardo cuando el usuario se encuentra haciendo *roaming*. La señalización necesita pasar por el operador en el cual el usuario hizo su suscripción incluso cuando fuera posible llegar al otro extremo de la comunicación por un camino más rápido. La red propietaria del usuario adquiere así una visión completa de las acciones del usuario y el control sobre ellas. Un retardo añadido es la consecuencia inmediata asociada con esta restricción. Sin embargo, este retardo solo afecta a la señalización y no a la sesión subyacente, así que en improbable que tenga un impacto significativo.

Las redes intermedias no necesitan almacenar datos globales de la sesión. Por lo tanto, estas redes intermedias deberían recopilar únicamente información a cerca de los recursos locales asociados a cada sesión. El servidor AAA de la red de acceso a la que está conectado el usuario, debe guardar un CDR de los parámetros globales de la sesión. De esta forma, toda red involucrada en la comunicación posee la información necesaria para poder facturar al operador con el que el usuario hizo la suscripción.

Como se demuestra en la sección 10.2.4, la solución propuesta es este documento es escalable y se puede aplicar en redes de cualquier dimensión.

Este proyecto de investigación es el resultado de mi trabajo durante una beca de 6 meses en Ericsson.

Abstract

With the increasing shift to the Internet Protocol [3] for all networks and the desire of telecommunications service providers to offer new value to their customers, the need exists to coordinate the delivery of end-to-end quality of service so that providers may offer new services to support their customer's applications. The key objective of the EuQoS project is to research, integrate, test, validate, and demonstrate end-to-end QoS technologies to support advanced QoS-aware applications over multiple, heterogeneous research, scientific, and industrial network domains. End-to-end quality of service support for multiple applications is a great added value and could become the next major growth spurt in the telecommunications industry.

A crucial requirement for the QoS model is that it must not add significant complexity to the existing mechanisms and must be compatible with legacy applications and equipment. Proxy signaling handlers will be used to satisfy the latter constraint.

This thesis analyzes possible roaming scenarios and how Authentication, Authorization, and Accounting should be tackled. I seek to provide reasonable solutions and to consider the current environment, always trying to re-use, when possible, the existing architecture and components.

Sammanfattning

De huvudsakliga målen med EuQoS projektet är att integrera, testa, utvärdera och demonstrera QoS från en ende av kommunikationen till en annan för att stödja avancerade QoS tillämpningar inom multipla nätverksdomäner industri- och forskningsverksamhet. Denna nya teknik är av stort värde och kan komma att bli nästa stora steg i utvecklingen av telekommunikation. Användandet av IP-nätverk ökar och i samma takt ökar även efterfrågan av nya och bättre kommunikationstjänster. Därför finns det ett stort behov av QoS hos Internetleverantörerna som ständigt måste tillgodose kundernas önskemål.

Det är viktigt att QoS modellen inte komplicerar den redan existerande tekniken. Därför måste den vara kompatibel med befintliga tekniker och utrustning. För att uppfylla dessa krav kommer Proxy signalbehandlare att användas.

Den här rapporten behandlar möjliga roaming scenarier och hur AAA bör hanteras. Jag har som mål att presentera rimliga lösningar som tar hänsyn till miljön. För detta används i största möjliga mån redan existerande infrastruktur.

Acknowledgements

First of all, I would like to thank my master thesis advisor, Professor Gerald Q. Maguire Jr., for his assistance, patience, and for his inestimable comments.

I would also like to express my most sincere gratitude to my advisors at Ericsson España, Miguel Angel Recio, for his support, the excellent working environment, and also for interesting discussions with him, which have helped me during this period.

Thanks also to Xiaoying for her opposition and to Adam for her Swedish translation.

Finally, I want to take the opportunity and express my deepest regards to my parents, my brothers, my friends, and Loreto for the great love and for the strength they gave me.

Table of Contents

Table of Figures:

Table of Tables:

1 INTRODUCTION

1.1 General overview

The key objective of EuQoS [1] is to research, integrate, test, validate, and demonstrate end-to-end Quality of Services [2] (QoS) technologies to support an infrastructure upgrade for advanced QoS-aware applications - voice, video-conferencing, video-streaming, educational, tele-engineering, and medical applications - over multiple, heterogeneous research, scientific, and industrial network domains.

With regard to this thesis, the document is the result of and internship in Ericsson España. We have been asked to cooperate and develop the AAA (*Authentication, Authorization and Accounting)* [4] module within the EuQoS project. Therefore, I will focus my thesis on the application of **AAA in roaming environments**. However, first I will describe some general issues of the EuQoS system in order to give a global overview of the system.

The EuQoS system will support the delivery of end-to-end QoS. As QoS is primarily a challenge for the access network, the EuQoS system will be developed and tested on various types of research access networks together with the GEANT [5] core that provides Pan European backbone support. This heterogeneous infrastructure, which models the production networks of the future, requires a QoS technical solution that has not been synthesised to date. The EuQoS project will propose and develop new QoS mechanisms which build upon the state of the art and incorporate the following mechanisms: Monitoring and Measurements, Admission Control, Failure Management, Signaling & Service Negotiation, Security and AAA, Charging, and Traffic Engineering & Resource Optimisation.

I would like to mention that before I was able to write any single line of this document I have carried out an extensive literature search. I wanted to find what others have already done in this field, so that I don't reinvent the wheel.

Some of the information that is depicted in this document is compiled from the public project deliverables that were already provided in to the European Commission by the EuQoS project.

Even though I was assigned to a specific part of the project, it is necessary to know how everything works together in order to obtain better integration. The study of all project deliverables yielded to an overview of the whole system.

There is an enormous similarity between the architecture of this system and IP Multimedia Subsystem[1] (IMS). Thus, some of the concepts applied to develop a roaming solution for EuQoS take ideas from the book "The IMS IP Multimedia Concepts and Services in the Mobile Domain" [14].

[1] IMS stands for *IP Multimedia Subsystem*, a concept developed and specified by the 3rd Generation Partnership Project (3GPP) (http://www.3gpp.org/). More information can be found at: http://www.unstrung.com/document.asp?site=unstrung&doc_id=70823&page_number=1

In addition to the literature exposed in the references, Ericsson has its own internal Technical Reports, which have been consulted several times to clarify some concepts. Ericsson has its own IMS implementations, SIP presentations, and other protocol reports that helped along in the research.

1.2 Thesis Scope

This thesis analyzes possible roaming scenarios and how Authentication, Authorization, and Accounting should be tackled. I expect to contribute with some implementation proposals within the time bounds of my thesis project. The application of the system in the real world will be discussed later in the document. I seek to provide reasonable solutions and to consider the current environment, always trying to re-use, when possible, the existing architecture and components. Therefore the output of this work is not to provide an exact quantitative analysis to the area problem based on accurate numeric values, but providing a roaming theoretical solution for the EuQoS system. As a result, the conclusions of this work will constitute a background for future implementations and studies dealing with this topic.

1.3 Thesis Outline

The thesis report is divided into the following chapters:

- Chapter 1 gives introduction to the thesis.

- Chapter 2 is a general overview of Authentication, Authorization, and Accounting. The main issues concerning AAA, and an introduction to charging and data records are described in this section.

- Chapter 3 explains the Diameter protocol and compares it with RADIUS; advantages and drawbacks are commented upon in detail.

- Chapter 4 describes several Charging issues, as well as online and offline charging models.

- Chapter 5 gives an overview of SIP. The SIP terminology and methods are described in this chapter. SIP will be used in addition to Diameter for the communication between the AAA server and the proxy servers. SIP has also been chosen for the resource reservation, but an enhancement has to be made.

- Chapter 6 explains the most relevant issues concerning the session description protocol (SDP).

- Chapter 7 describes the EuQoS end-two-end architecture. A first approach to the EuQoS system is related here, including a mapping of the AAA scheme onto the EuQoS architecture.

- Chapter 8 is about the EuQoS use cases. This chapter starts with a roaming overview, which is followed by a general perspective of accounting in EuQoS.

- Chapters 9, 10, and 11 explain the use cases: a non-roaming scenario, a roaming scenario, and non-EuQoS roaming (respectively).

- Chapter 12 analyzes the design and gives the conclusion and future work.

- An appendix defines the main terms and clarifies their meaning in the context of this document; it also includes the abbreviations used in this thesis and includes a press release describing EuQoS.

2 Authentication, Authorization, and Accounting

AAA stands for Authentication, Authorization, and Accounting. Nowadays, many of the protocols used in the Internet do not provide any security. A malicious hacker can easily steal passwords from the network using “Sniffers”. Thus, applications that send clear text passwords (unencrypted) over the network are extremely exposed. Even worse, some applications rely on the client program to be “honest” about the identity of the user. Other applications rely on the client to restrict its activities to those that it is allowed to do, with no other verification by the server. This is why it is necessary to check the identity of the person or client you are communicating with, i.e. **authenticate** them. The authentication process confirms that a user who is requesting services is a valid user of the network services requested.

When the user agent is authenticated, next step is to determine what services to allow the requester. **Authorization** refers to the granting of specific types of service (including "no service") to a user, based on their authentication, what services they are requesting, and the current system state. Authorization is based on policy-based decisions and determines the nature of the service which is granted to a user.

Any Autonomous System (AS) wants to know what happens in its network. **Accounting** information is gathered in order to keep track of the consumption of network resources by users. To cover the increasing roaming and mobile subscriber, ISPs may choose to pool their network resources while keeping control over their subscribers access, usage, and billing information. This information may be used for management, planning, billing, or other purposes. Accounting requires coordination between various autonomous systems supported by the ISPs in partnership with each other. *Real-time accounting* means that the accounting information is delivered concurrently with the consumption of the resources, while *batch accounting* refers to accounting information that is saved and delivered at a later time. The identity of the user, the nature of the service delivered, when the service began, and when it ended are part of the typical information that is gathered in accounting.

The purpose of AAA is to meet the above challenges in a simplified and scalable way. AAA defines a framework [26] for coordinating these individual disciplines across multiple network technologies and platforms [27].

Mobility is a very important component influencing AAA. The goal is to achieve the capacity of dynamically assign a mobility anchoring point in either the home or foreign network, as well as distributing the session keys to these mobility agents. This is not feasible with RADIUS and Mobile IP since that keys cannot be distributed dynamically, and thus must be pre-established. That is why the AAA and Mobile IP IETF Working Groups have defined the interaction between Mobile IP and Diameter to support this functionality.

In today’s world, the huge acceptance of mobile devices creates the need of using the terminal to access a telecom service from anywhere independently of their current location. Thus, a user should be able to access to resources being provided by an administrative domain different than their home domain (which will be referred as *foreign domain*). Services from a foreign domain require, Authorization, which leads directly to Authentication, and of course Accounting.

The AAA system in a foreign domain is likely to request or require the client to provide credentials which can be authenticated before access to resources is permitted.

Mobile IP is a technology that allows a network node to migrate from its "home" network to other networks, either within the same administrative domain, or to other administrative domains. The formal description of Mobile IP is detailed within [28][29][30][31]. Mobility between different domains, which require AAA services, creates a demand to design and specify AAA protocols.

However, this implies that the correspondent host must keep track of the mobile's care-of-address and home agent, and that the old foreign agent should forward packets to the new foreign agent.

A major design goal of a wireless network must be to provide a durable IP address to the user's mobile node, which persists even when the user moves from cell to cell in the wireless provider's network.

Beyond providing the functionality described above, to support a 3G infrastructure a AAA server should permit a wireless provider to deliver a fast, high-quality wireless Internet service that subscribers demand, as economical as possible. To achieve this, the AAA server must [35]:

- Be able to **handle a transaction volume significantly larger** than that of a wired communication. Within a 3G/wireless Internet session, the AAA server must process authentication, authorization, and accounting transactions when the subscriber first logs-in, and also every time the subscriber moves between coverage zones. This requires that AAA server can support transaction rates in the thousands-per-second range (depends on the number of subscribers connected to the network in a given time).

- **Integrate seamlessly with the provider's network infrastructure**. AAA server must support network access servers from the widest possible range of vendors to minimize the provider's administrative overhead and leverage existing investments. The AAA scheme should integrate with the existing provisioning and billing systems.

- **Be easily scalable.** AAA server should enable redundant access to authentication, authorization, and accounting systems, and should easily scale without sacrificing performance.

- **Support advanced proxy capabilities and policies** to manage wholesaling and roaming agreements.

More information about the functional and performance requirements that Mobile IP places on AAA protocols can be found in [32][33], where also some related AAA models are exhibited.

When a mobile node moves between two foreign networks, it has to be re-authenticated. If the home network has both multiple Authentication, Authorization, and Accounting

(AAA) servers and Home Agents (HAs) in use, the Home AAA server may not have sufficient information to process the re-authentication correctly. The Home AAA server needs to know the identity of the HA that is using the mobile node in order to forward the request to the correct HA.

In [34] a Mobile IP extension is defined. The extension carries identities for the Home AAA and HA servers in the form of Network Access Identifiers (NAIs). This extension allows a HA to pass its identity (and that of the Home AAA server) to the mobile node, which can then forward it on to the local AAA server when changing its point of attachment.

Redundancy can sometimes be beneficial when building networks. One might place multiple AAA servers in one domain to achieve this redundancy. If a user registers via a visited network, the authentication request has to be sent to the Home domain since one of the AAA servers in the home domain will handle the request. At a later point, if the user moves to another domain different than home, the User Agent (UA) [57] has to be authenticated again. However, due to the redundancy offered by the AAA protocol, it can not be guaranteed that the authentication will be handled by the same AAA server at the home domain as the previous one, which can cause problems when trying to contact the HA assigned during the session. The Mobile IP extension can be used to solve this problem. As it is explained in [28], the home agent must include the AAAH NAI in the registration reply message, sent via the AAA infrastructure, which the mobile node then MUST include in every subsequent registration request sent to a foreign agent when changing point of attachment.

Furthermore, the only information that is normally available about the home agent in the registration request is the IP [3] address as defined in Request For Comments[2] (RFC) 3344 [28]. On the other hand, this may not be enough since some AAA protocols such as Diameter [24] use realm based routing; such a AAA infrastructure needs to know the Fully Qualified Domain Name (FQDN) of the HA to be able to correctly handle the assignment of the HA. A reverse DNS lookup would only reveal the identity of the Mobile IP interface for that HA IP address, which may or may not have correspondence with the home agent FQDN identity.

One way of solving this problem would be for the home agent to also include its own identity in the registration reply so that it can be included by the mobile node in the coming registration requests when changing point of attachment [28].

The interaction between Authentication, Authorization, and Accounting (AAA) systems and the Quality of Service (QoS) infrastructure is to become a must in the near future [18]. This interaction will allow rich control and management of both users and networks. DIAMETER and DiffServ are likely to turn into the future standards in AAA and QoS systems, but they are not designed to interact with each other. To face this, in “Mechanisms for AAA and QoS Interaction” [18] they propose a new Diameter-Diffserv interaction model and describe the Application Specific Module (ASM) implemented to allow this interaction.

[2] RFC (Request For Comments) is an Internet information document or standard. RFCs provide the technical details that describe a protocol.

To finish with, it is important to remark that to cash in on the huge opportunity of today's telecommunications, service providers need an AAA solution that combines the necessary authentication, Mobile IP, service delivery, and accounting technology with the raw performance, ease of integration, manageability, and scalability that guarantees the fastest and highest return on their infrastructure investment.

2.1 Authentication

Authentication is the process by which a computer, computer program, or another user attempts to confirm that the computer, computer program, or user from whom the second party has received some communication is, or is not, the claimed first party, i.e. authentication is the process of determining whether someone or something is, in fact, who or what it is declared to be [36].

Authentication defines the verification of the identity of a subject. Authentication mechanisms can be classified as follows [15]:

- *Knowledge*-based authentication founds on the knowledge of shared secrets, such as PINs (Personal Identification Number) and passwords.

- *Cryptography*-based authentication includes digital signatures, challenge-response mechanisms, and message authentication codes. The user owns a private key as a characteristic.

- Authentication based on *biometrics* uses inherent information on subjects like fingerprint, voice, and eye characteristic.

- Authentication based on *secure tokens* binds the subject to some kind of ownership, e.g. the ownership of a smart card. It is combined mostly with cryptographic mechanisms to transfer the information on the token to the authenticator.

- *Digitized signatures*, including digital images of handwritten signatures and signature dynamics (i.e., measurements of the direction, pressure, speed, and other attributes of a handwritten signature) are not widely used so far.

The AAA server compares the user-supplied authentication data with the user-associated data stored in its database, and if the credentials match, the user is granted network access. A mismatch results in an authentication failure and a denial of network access. An authentication policy describes whether authentication has to be done and which authentication mechanisms and algorithms (actions) should be used under which constraints.

Authentication is commonly done through the use of logon passwords. Knowledge of the password is assumed to guarantee that the user is authentic. Each user registers initially (or is registered by someone else), using an assigned or self-declared password. On each subsequent use, the user must know and use the previously declared password. The weakness in this system for transactions that are significant (such as the exchange of money) is that passwords can often be stolen, accidentally revealed, or forgotten.

For this reason, Internet business and many other transactions require a more stringent authentication process. The use of digital certificates issued and verified by a Certificate Authority (CA) as part of a public key infrastructure is considered likely to become the standard way to perform authentication on the Internet.

2.2 Authorization

Authorization is the process of determining, by evaluating applicable access control information, whether a subject is allowed to have the specified types of access to a particular resource. Usually, authorization is in the context of authentication. Once a subject is authenticated, it may be authorized to perform different types of access. Authorization mechanisms can be categorized in two major classes [15]:

- *Authentication-based* mechanisms require an authentication of the subject as precondition for the authorization. The information for the authorization decision is stored at object systems, such as in Access Control Lists (ACLs) of operating systems in the form “User S is allowed to perform action A on an object O”.

- *Credential-based* mechanisms use credentials which are trustworthy information being hold by subjects of an authorization process. Credential-based mechanisms are widely accepted in E-Business. Authorization policies define those actions a subject is permitted to perform on an object. An authorization policy may be positive (permitting) or negative (prohibiting).

There exists obviously a great similarity between policies and mechanisms for authentication-based authorization. For credential-based mechanisms a credential has a similar form as a policy, whereby the set of objects has only one element which is the user (may be anonymized) who owns the credential.

Authorization defines what rights and services the end user is allowed once network access is granted. This may include providing a *user profile* to determine which applications or protocols are supported. Authentication and authorization are usually performed together in an AAA-managed environment.

2.3 Accounting

- IETF: Accounting is the act of collecting information on resource consumption data for the purpose of trend analysis, capacity planning, auditing, billing, or cost allocation.

- 3GPP: Accounting is the process of apportioning charges between the home environment, serving network and user.

Accounting management requires that resource consumption be measured, rated, assigned, and communicated between appropriate parties [37].

Accounting provides the methodology for collecting information about the end user's resource consumption, which can then be processed for billing, auditing, capacity-planning purposes and also for abuse handling purposes in order to monitor and act against malicious users [38].

An accounting system takes two major tasks [15]: to collect data from metering systems and to distribute data to users of accounting records. Therefore, two kinds of policies belong to the collection and distribution.

For the collection task a metering policy describes which information has to be metered by a metering system and transported to the accounting system. These policies are event triggered by a signalling event unless static meters are used, which collect data for all flows in a fixed granularity.

The user of accounting records can, depending on his objective, specify via an accounting policy, which information he needs at which time from the accounting system. This policy can be event triggered by internal events, the billing system request on an accounting record, or by external events like the end of month. Policies can be obligation driven also, i.e. if a new charging scheme is placed, then new accounting information has to be collected.

Since accounting applications do not have uniform security and reliability requirements [37], it is not possible to devise a single accounting protocol and set of security services that will meet all needs. [37] Describes the currently available tools that can be used to meet the requirements of each application as well as the state of the art in accounting protocol design.

An extensible classification scheme for AAA Accounting Attributes is proposed in [39], where several IETF and ITU-T documents related to Accounting are summarised.

Many existing accounting record formats and protocols as TIPHON [40] and RADIUS Accounting [41] are of limited use due to their single-service descriptive facilities and lack of extensibility. While some record formats and protocols support extensible attributes (like RADIUS Accounting [41] none provide identification, type checking, or versioning support for defined groupings of attributes (service definitions). Advantages and disadvantages of integrated versus separate record formats and transport protocols are also discussed in [39].

3 AAA protocols: RADIUS and DIAMETER

Internet services providers, corporations, and others providing remote services have to face authentication, service delivery, and billing issues daily. Some time ago, they turned to solutions based on *Remote Authentication Dial-In User Service (RADIUS)* [22], a protocol developed and supported by a working group within the Internet Engineering Task Force (IETF) that describes the communication between network access devices and a server for AAA purposes.

Historically, the RADIUS protocol has been used to provide AAA services for dial-up PPP (Point-to-Point Protocol) [23] and terminal server access. Over time, as routers and network access servers (NAS) increased in complexity and density and with the arrival of new services, the RADIUS protocol has become increasingly unsuitable for use in such networks [38]. These changes, combined with a massive deployment of the RADIUS protocol have uncovered some fundamental issues that will be addressed in next section.

All signs points out to our heading towards a wireless era. The development of Mobile IP [28], and it is recently rising popularity, has caused a greater number of ISPs to see the benefits of an AAA protocol being able to interact with the Mobile IP protocol. Most of the new services such as Voice over IP, Fax over IP, and Mobile IP require similar functions to authenticate, retrieve authorization information, and generate accounting records for billing purposes. If each service creates its own protocol to achieve this, this requires customers to deploy several different policy servers, which increases the cost of administration and complicates the deployment of multiple services.

The Internet Engineering Task Force is in the process of standardizing a new Authentication, Authorization, and Accounting protocol called Diameter, which will replace RADIUS, the legacy AAA protocol.

Diameter offers a general answer to the above stated situations implementing. The base protocol [24], which defines header formats, security extensions, and requirements as well as a small number of mandatory commands and attribute value pairs (AVPs). The Diameter base protocol can be extended to support new functionality. This allows each Working Group within IETF to use Diameter; while adding their new service specific requirements in a new Diameter extension.

3.1 RADIUS drawbacks

The RADIUS protocol was developed in the early 1990's to provide scalability for dial-in PPP and telnet servers. Since then, networks have become more complex (e.g. adding roaming) and the Network Access Servers have increased in complexity and density. According to [45], where a detailed statement of each issue can be found, analysis of the RADIUS protocol uncovered the following fundamental matters that needed to be fixed:

- Strict limitation of attribute data
- Strict limitation on concurrent pending messages
- Inability to control flow to servers
- No retransmission procedure
- End to end message acknowledgment
- Limited server failure detection
- Silent discarding of packets
- Inefficient Server Fail-Over
- Inefficient use of RADIUS servers in proxy environments
- No unsolicited messages
- Replay Attacks
- Hop-by-Hop security
- No support for vendor-specific commands
- No alignment requirements
- Mandatory Shared Secret

The RADIUS protocol, and its associated extensions, is presently not fully compliant with the AAA Network Access requirements. However, it is possible with a small effort to extend present procedures to meet the requirements as listed in, while maintaining a high level of interoperability with the wide deployment and installed base of RADIUS clients and servers [46].

3.2 DIAMETER

The Diameter protocol was designed as a next generation RADIUS protocol. Diameter was not created out of nothing; it contains the basis RADIUS format and is designed with roaming and high density NASes in mind.

The Diameter architecture consists of a base protocol [24] and a set of applications. The idea of Diameter is to create a base protocol which easily can be extended in order to allow new access methods. Common functionality to all supported services is implemented in the base protocol, while application-specific functionality may be provided through the extension mechanism. The base protocol must be supported for all Diameter applications, and defines the basic Diameter message format, a few primitives and the essential security services offered by the protocol.

For several years the question as to whether RADIUS should operate over UDP or TCP has led to intense discussion. For Diameter, UDP has been summarily dismissed since it would require more logic in the application layer. Now the debate has moved to the question of using SCTP or TCP. One of the shortcomings of TCP is the lack of a quick retransmission and fail-over scheme, which by contrast, are supported in SCTP. This capability is a requirement for the Diameter protocol, which must be able to operate over a transport protocol that has an aggressive retransmission strategy in order to efficiently switch to an alternate host when the peer in question is no longer reachable. With this in mind, the latest drafts have prescribed SCTP as the transport layer protocol to be used. It should be noted however, that while SCTP is generally regarded as the most complete technical solution, the working group is still engrossed in political arguments that continue to plead for TCP.

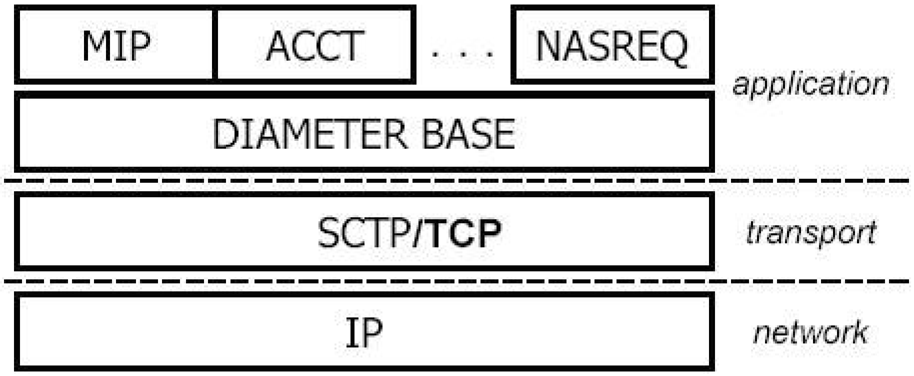


Figure 1: The Diameter protocol stack

The base protocol is also capable of running over IPv4 or IPv6 networks, and is capable of distinguishing between IPv4 and IPv6 addresses. The basic RADIUS model was retained while fixing the associated weaknesses in the protocol. Diameter does not share a common protocol data unit (PDU) with RADIUS, but does borrow sufficiently from the protocol to ease migration.

The Diameter base protocol itself is able to determine how messages are sent, negotiate capabilities, and determine how peers may eventually be abandoned. The base protocol also defines certain rules which apply to all exchanges of messages between Diameter nodes.

The base protocol is a session-oriented protocol based on a peer-to-peer communication model, as opposed to a client-server model. The following goals have motivated the design of the base protocol [45]:

- Lightweight and simple to implement protocol
- Large Attribute Value Pair (AVP) space
- Efficient encoding of attributes, similar to RADIUS
- Support for vendor specific AVPs and commands
- No silent discarding of messages
- Support of unsolicited messages
- Integrity and confidentiality at the AVP level
- Better hop-by-hop security than RADIUS
- One session per authentication/authorization flow
- Support for large number of simultaneous pending requests
- Reliability and well-defined fail-over scheme provided by underlying SCTP
- Ability to quickly detect unreachable peers
- Provide redirect (referral) services, to allow bypassing of proxies when appropriate.

The Diameter base protocol is thought to simply provide a secure transport for the messages defined in the various application-specific extensions. It is therefore essential that the base is lightweight and simple to implement.

Information is encapsulated within an Attribute Value Pair (AVP). Different extensions to the base protocol allow the usage of different access technologies, by defining special command codes and AVPs. The NASREQ extension [47] has been carefully designed to ease the burden of protocol conversion between RADIUS and Diameter, support RADIUS authentication protocols, PPP Extensible Authentication protocol (RFC 2284 EAP) [48], and authorization as needed by NAS-Services. Mobile IP extensions define AVPs to support Mobile IP across disparate administrative domains [49]. The Diameter base protocol provides an AAA framework for Mobile-IP, NASREQ, and ROAMOPS (ROAMing OPerationS) between others. Using these, a Diameter server is able to authenticate, authorize, and collect accounting information for services requested by a mobile node. The accounting extension [50] defines a set of generic accounting AVPs that can be used for all services and supports real-time accounting. Each Diameter extension defines its own service specific accounting AVPs.

A unique AVP Identifier is assigned to all data objects in order to be able to distinguish the data contained. An AVP consists of three parts: the Identifier (AVP-Code), Length of the data and the Data. The AVP Identifier namespace must be sufficiently large to ensure that future protocol extensibility is not limited by the size of the namespace, as occurred with the RADIUS protocol. Furthermore, vendors wishing to add proprietary extensions must be allowed to do so by using a vendor-specific namespace, managed by the Internet Assigned Numbers Authority (IANA).

<table>
<tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr>
<tr><td colspan="32">AVP - CODE</td></tr>
<tr><td colspan="16">AVP Length</td><td colspan="11">reserved</td><td>P</td><td>r</td><td>V</td><td>r</td><td>M</td></tr>
<tr><td colspan="32">Optional Vendor - ID</td></tr>
<tr><td colspan="32">Data…</td></tr>
</table>

Figure 2: Diameter AVP header

Where,

P = protected

r = reserved

V = vendor-defined AVP

M = mandatory

With the exception of a few security-related errors, the Diameter protocol requires that all messages be acknowledged. This could be either with a successful response or one that contains an error code. While the RADIUS protocol is client-server based, the Diameter protocol is peer to peer, allowing unsolicited messages to be sent to mobility agents (e.g. NASs, home/foreign agents, etc). There are many benefits to peer-to-peer AAA protocols, some of which include on-demand retrieval of accounting data, and server-initiated session termination.

<table>
<tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr>
<tr><td colspan="10">reserved</td><td>E</td><td>I</td><td>R</td><td colspan="3">Ver</td><td colspan="16">Message length</td></tr>
<tr><td colspan="32">Hop-by-hop Identifier</td></tr>
<tr><td colspan="32">End-to-end Identifier</td></tr>
<tr><td colspan="32">Vendor-ID</td></tr>
</table>

Figure 3: Diameter message header

Where,

E = Reply expected
I = Interrogation
R = Response

The Diameter base protocol provides for hop-by-hop security, similar to the scheme employed by RADIUS today. However, the Diameter protocol also provides for replay protection through a timestamp mechanism. This security scheme requires a long-lived security association to be established by peers, or can make use of keying material negotiated out of band. The CMS draft [53] explores the idea of providing end-to-end security, but this work is currently abandoned.

Additionally, Diameter specifies Internet Protocol Security (IPSec) [51] and Transport Layer Security (TLS) [52] for securing communication between Diameter nodes, where IPSec is suggested to be used primarily for intra-domain exchanges and TLS for protection of inter-domain communication. In environments where there is not a trusted third party agent, end-to-end security is needed. The Base Protocol also allows the built-in security measure to be turned off (i.e. in cases where IPsec is in use).

The Diameter protocol is a session-oriented protocol, meaning that for each user being authenticated, there exists a session between the initiator of the authentication/authorization request and the home Diameter server. Sessions are identified through a session identifier, which is globally unique at any given time. All subsequent Diameter transactions (e.g. accounting) must include the session identifier to reference the session. A Session termination message exists in order to end a Diameter session, and all sessions have a timeout value in order to ensure that they can be cleaned up properly.

3.2.1 Realm-based routing

Routing of Diameter messages are performed on a hop-by-hop basis in a manner analogous to the usage of DNS. Each Diameter server maintains a local realm-based routing table to assist in determining the server to forward a request. Each realm listed in the routing table indicates a server or list of servers, as well as the Diameter applications the servers support (as advertised during the capabilities negotiation). This could be useful in the event that a request should be routed to a particular server within a domain, based on which Diameter application originated the request.

When a server receives a request destined for another domain, it inspects its routing table to determine to which server to forward the request. Default servers can also be configured in the routing table to handle requests with no specific matches.

3.3 Radius vs Diameter

The main reasons of developing the Diameter protocol were mentioned in the previous section. In spite that all of them should be taken into account, scalability and the need of the provision of an AAA protocol able to operate with roaming facilities are the most relevant.

With regard to scalability, RADIUS is not scaleable since the RADIUS protocol states that the identifier field, found within the header, is used to identify transmissions. This identifier field is only one byte long; therefore, the number of requests that can be pending simultaneously is only 255.

The Diameter base protocol has four bytes identifiers, which make it scaleable since it and can support up to 2^{32} (one byte = 8 bits) requests at the same time.

While RADIUS deals with start, stop, and activity data including various accounting, tunneling, and general attributes, Diameter inherits all of them and defines a secure protocol suitable for a roaming environment to transfer these accounting attributes.

The following paragraphs show more aspects in where Diameter overcomes RADIUS [38]:

3.3.1 Authentication

- Authorization without Authentication

The Radius protocol does not support non-Authenticated Authorization, because the protocol does require some form of credentials in request messages. The Diameter protocol does not require Authentication information to be included in the request to the other peer.

- Replay attacks and denial of service attacks

RADIUS does not contain end-to-end Authentication just hop-by-hop authentication, the protocol does not include any replay attack prevention. The Diameter protocol prevents replay protection through a timestamp mechanism and through the support of end-to-end Authentication.

3.3.2 Authorization

- RADIUS gateway capability

AAA protocols need to have RADIUS capabilities in order to ease migration and be able to interact with the major AAA agents today, which maybe neither needs nor wants to migrate to Next Generations AAA protocol. The Diameter protocol was created with RADIUS capabilities in mind,

- State-Reconciliation

Diameter on the other hand through the former resource management application and now by the base protocol do support the messages needed for state recovery and therefore supports State-Reconciliation.

3.3.3 Accounting

- Support of unsolicited messages

Since Diameter is a session based protocol (peer-to-peer) it supports unsolicited messages from the Diameter “server” (any direction from peer to peer) in traditional “server to client” sense. In comparison RADIUS does not support unsolicited messages since it is a client/server protocol that requires a client to initiate a request. Support of Unsolicited messages is typically needed for accounting purposes, to request that a NAS terminate a specific user session and to support of services where session/configuration information have to be changed during a session.

3.4 Future of the Protocol

The IETF AAA Working Group has been working over the past couple of years to develop a second generation AAA protocol to succeed the original AAA protocol, RADIUS. This has been necessary due to RADIUS's enormous complexities when deployed in large scale networks, as well as its unsuitability in upcoming 3G cellular networks.

This replacement, Diameter, was designed specifically to correct the problems that plagued RADIUS, as well as providing new functionality and greater extensibility to support future networks as well. The protocol is in the final stages of standardization, and once complete, Diameter will be poised to spring up in all types of networks, such as those deployed by 3G cellular operators, ISPs and corporate networks.

The basic concept behind Diameter is to provide a base protocol that can be extended through Diameter Applications in order to provide AAA services to new access technologies. Currently, the IETF AAA working group is only concerned with Internet access, both in the traditional PPP sense as well as taking into account the ROAMOPS model, and Mobile IP. Although Diameter could be used to solve a wider set of AAA problems, the IETF working group has limited the scope of the protocol in order to ensure that efforts remain focused on satisfying the requirements of network access.

After the initial applications (NASREQ, MIP, CMS, along with the base protocol) have become standardized, the AAA working group will open its charter to focus on new Diameter applications. A number of such applications have already been proposed today, such as a MIPv6 Application, which defines how Diameter will support Mobile IPv6, and a Multimedia Application, which allows Diameter to be used in supporting SIP for multimedia session establishment.

4 Charging

Charging applies business and policy-based decisions to accounting data to produce bills. While a charging policy defines tariffs and parameters, which are applied by charging mechanisms, a charging mechanism provides the infrastructure to calculate final charges for service usage based on accounting information. Assuming that a suitable overall business policy exists, a specific policy can enable ISPs to gain income and potentially survive in the market.

More commonly, the word charging simply means the process of debiting an account. The processing of the charging involves Charging Control; this normally includes a process called “rating” i.e., computing a price. Serving Elements generally rely on the rating functionality of Charging Control; hence they need not have their own rating functionality.

Additionally, it is important to remember that fulfilling charging duties involves more than simply implementing an interface, it involves making a system and the services it delivers a key part of an operator’s business process; charging is more than a checklist item – it is a key process.

Currently, EuQoS does not define a fixed charging model. Several potential options are commented upon this chapter.

4.1 Why Charging?

When users sign a contract with an operator, the agreement specifies that the operator will offer services to this subscriber and as the subscribers find these services valuable, they are willing to pay for them based on the specifications in the contract. The simplest way to charge a subscriber is by a periodic subscription fee, allowing them to use all they wish. This method is not fair; since more active users might have paid more since they generate more traffic, and the less enthusiastic subscriber would think it was expensive and not sign up. Therefore, a widely accepted pricing model is to let the subscribers pay for their actual consumption.

By supporting charging in the way described in this chapter, the EuQoS system will be ready for integration into the operator’s charging environment, and also allow the operator a greater freedom to implement any business model that would maximize profits. The operator can freely implement any competitive tariffs, discounts, and promotions without being restrained by inflexible design.

4.1.1 Serving Elements

The term Serving Element refers to the role that any system assumes when it supports charging. As a Serving Element, the system will interact with the Charging Control. The charging client is the part of a Serving Element responsible for interacting with Charging Control.

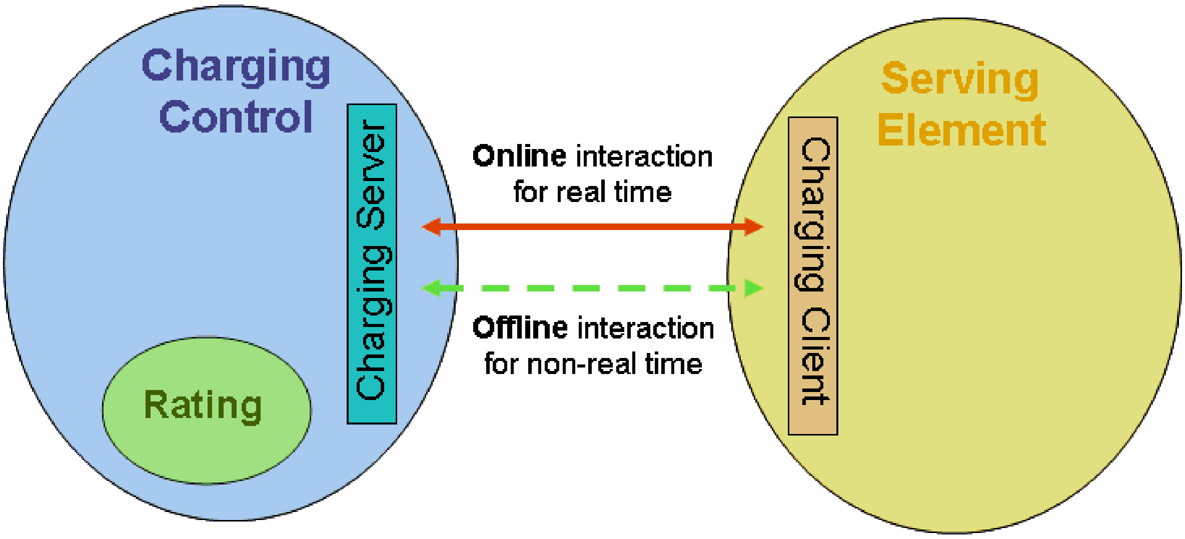

Charging Control
Charging Server
Rating
Online interaction for real time
Offline interaction for non-real time
Charging Client
Serving Element

Figure 4: Serving Element and Charging Control interaction

4.2 On-line charging

In on-line charging the credit that covers the cost for usage is granted before any further resources are allocated to or consumed by the subscriber. This assures that all consumption is supported by credit that the subscriber has with the operator. In the case of a **prepaid** subscriber, such credit corresponds to the prepaid amount remaining on the account, and in case of a **postpaid** subscriber, the credit corresponds to the risk the operator wants to take with this particular subscriber. To the operator this is **Credit Control**. For the subscriber this is **Spending Control**; a guarantee that she will have no liabilities beyond the credit.

An on-line charging mechanism requires interaction between Serving Elements and Charging Control in real-time. This ensures effective credit supervision and reduces service-rendering delays. These delays may occur when the Serving Element must wait for an answer from Charging Control.

4.2.1 Serving Element duties

On-line charging is a mechanism which requires certain behavior of the Serving Element.

1 The Serving Element asks the Charging Control before service rendering to grant enough credit. Depending on the requested service, this needed amount of credit can be, for instance, the cost of the service establishment plus the cost of the first minute (in case of time based charging).

2 The Serving Element manages the reserved credit as follows:

- increases used service units every time a service deliver is completed (e.g. adds one to a counter with every MMS sent)
- requests new credit if used credit is close to granted credit and service rendering needs to continue
- informs about consumed credit to Charging Control
- tells Charging Control to end charging when user stops the service
- carries out instructions received by Charging Control (e.g. stop service rendering) when there is no more credit available.

Note: All Serving Elements must support on-line charging.

4.2.2 Charging Control duties

On-line charging is a mechanism which requires certain behavior of Charging Control:

1. Checks that the subscriber account is still valid, i.e. that it has not expired because account has no credit or is closed.
2. Is in charge of check for any bonus points, promotional coupon, etc. that it could be used to pay for the service.
3. Translates credit requests containing some type of service units (i.e. one minute call) into a price for the particular subscriber.
4. Deducts used credit from account.
5. Tells Serving Element to end charged session when no more credit is available.

4.3 Offline charging

Offline charging, also known as non-realtime charging, is the traditional way of charging. The Serving Element reports service usage to Charging Control while (or after) service rendering. This implies that credit control takes place after the service has started or even after it has finished. Therefore the risk for the operator of not getting money for the service he provided to the user increases substantially. Thus on-line charging is the preferred mechanism since it eliminates such a risk.

However, offline charging has still its reasons to exist. The **on-line** charging request processing requirements of realtime charging **limits** the amount of **information** that can be included in such a request. Offline charging is not as time critical as on-line charging, and therefore more detailed information can be included. This detailed information is not restricted to data serving as input to charging, i.e. offline charging could be used to report detailed information on service usage to business support systems, as well as for monthly/quarterly billing, planning upgrades (resource dimensioning), input for fraud detection [54].

Offline charging input describes how a user has used a service, i.e. it is historical information. Even if an offline charging functionality of Charging Control is temporarily unavailable there is no harm because the Serving Element can buffer the information and send it to Charging Control when the service is again available.

Some operators have a charging solution in place, but do not yet support on-line charging. EuQoS Serving Elements have to adapt to this situation and support offline charging.

These characteristics suggest that offline charging is necessary as secondary charging mechanism, hence it must also be supported by Serving Elements.

4.3.1 The need for On-line and Offline Charging

We need to support **both** mechanisms because they are complementing and not simply competing: on-line charging offers spending control to the subscriber and credit control to the operator. The real-time nature implicitly limits the amount of information transmitted through a link, which means that an on-line process is not suitable for reporting service usage details not related to price determination or account debiting.

Charging is related to collecting money; hence failures of charging functionality may mean financial losses for operators. Therefore it is extremely important that charging works well even in tricky situations. Hence use of online and offline charging simultaneously is a must.

4.4 Charging process

Let’s consider a user that wants to watch a football game, which costs 1€. Charging for the match in this case involves:

1. First of all, the football match provider receives a request from a subscriber for viewing the game. Although, before the game is delivered to the user, the system asks Charging Control if the subscriber has enough credit for viewing it.
2. Charging Control calculates the cost for watching the match, rating, which is 1€, reserves this amount from the subscriber's account, and answers the system that the subscriber has enough credit.
3. The system delivers the football match (perhaps a stream). It creates a Call Detail Record (CDR) in which it stores subscriber related information (like user identity, user location, terminal identity, and class), details on the music video selected, the service delivery start, system load, etc.
4. After the video has been successfully delivered to the subscriber the system reports so to Charging Control and ends the charging session.
5. Charging Control deducts the reserved 1€ from the subscriber’s account and tells the system that the charging session has successfully finished.
6. The system updates the earlier created CDR. This CDR is stored in a specific directory from where it can be sent to Charging Control later on.
7. After some time (probably hours) all new CDRs are sent to Charging Control.

If the content provider was not able to deliver the complete video to the subscriber, then the Charging Control should be informed. In this case, the Charging Control must unreserved the 1€ and the subscriber will not be charged. The CDR created will contain detailed information about why the delivery has not been successful.

4.5 On-line Charging Protocol

The DIAMETER protocol was specified by IETF in RFC 3588 [24]. This is the base protocol for Authentication, Authorization, and Accounting (AAA). Diameter runs over TCP or SCTP. Diameter is an evolution of the RADIUS protocol, which is a well established AAA protocol, and therefore preferred by the charging departments of many providers. Diameter is scalable, and its functionality includes advanced recovery, redundancy, and load balancing. It is capable of 600 transactions a second, which is a big improvement over the 20 or so transactions a second possible with remote procedure call based (RPC-based) Multimedia Messaging Service (MMS) charging (see section 3.2 for more information about Diameter).

The base protocol is the bearer for application protocols, such as the **Diameter Credit Control Application (DCC)**, RFC 4006 [66]. This protocol (DCC) has been chosen by 3GPP as the realtime charging protocol for IMS [67]. This will be a strong driver for all charging systems and Serving Elements to support Diameter. A similar decision is being assessed by 3GPP2. Therefore, it is advisable for the EuQoS system to also use the DCC protocol when implementing on-line charging.

From a Serving Element developer point of view, DCC is recommended when building new real time charging interfaces. DCC is emerging as the primary (if not only) protocol for on-line charging. Today, most charging systems may not speak DCC, however I believe that they will have to sooner or later. Thus my recommendation is that EuQoS should support DCC as the main charging protocol.

4.6 On-line Charging Methods

There are basically three on-line charging methods: event charging, direct debiting, and session charging. Note that any system should concurrently do offline charging.

4.6.1 Event charging

If the credit that corresponds to the service has not been reserved, then service delivery will not be initiated. Note that if credit is reserved, then this reserved amount is unavailable to other services. Once the service has been successfully delivered, the reserved amount is deducted. On a delivery failure, the Serving Element reports to the Charging Control that no service was delivered and the reserved amount will once again be fully available in the account.

Note: Charging Control assigns a lifetime to the reservation. The Charging Control cannot wait forever for the system to confirm delivery and conclude the transaction. Once the reservation lifetime expires, the Charging Control assumes that the Serving Element session has been aborted. In that case the Charging Control shall assume that the service was not delivered, and refund the credit reservation to the account.

4.6.2 Direct Debiting

Direct debiting is a good option when delivery success is guaranteed. Under this condition, the requesting and reporting can be done in one transaction and there is no credit reservation phase. Example: the subscriber donates 1€ to charity. Here we have "no" service delivery, and thus the service delivery is always successful, i.e. it does not make sense to use any reservation based charging method.

Note: there are some services which may not generate a charging record for the customer, such as a call to 112. In such a case, the service is not charged. Therefore it only involves accounting, but not charging. The CDR will be gathered for accounting purposes.

4.6.3 Session Charging

The final consumed amount of resources given to a certain service are unknown at the start of the communication (e.g.., for an ordinary phone call the duration is not known beforehand, hence a fixed charge can not be computed). Credit for a **predicted consumption** is reserved for the delivery of the service. If the session continues and threatens to consume more than the granted amount, then the Serving Element shall request an additional credit amount (preferably with some remaining margin covering the gap before the new service has been granted). With each of these renewal requests, the Serving Element reports how much was consumed in the previous interval. In this case, the charging system rates the reported consumption, deducts its actual cost, and releases the previously reserved remaining credit.

Note that reservation time is supervised; the Serving Element cannot wait forever to continue the transaction. See also the note at event charging.

4.7 Interface requirements

4.7.1 On-line Interface

An on-line charging interface must be supported by all Serving Elements generating accounting events, given that the Serving Element can provide Charging Input Parameters that affect the subscriber charges.

Protocol to be used:

- DIAMETER Credit Control Application

4.7.2 Offline Interfaces

One offline charging protocol shall be supported by all Serving Elements, which generate accounting events.

Protocol to be used:

- Diameter with an accounting application (Standardized, i.e. NASREQ, Multimedia or vendor specific according to DIAMETER base protocol) to transfer CDRs. The DIAMETER protocol/framework can handle roaming or TAP procedures as used for WLAN billing [68], and can be used in the offline case. Another valid option is to transfer CDRs using FTP.

4.8 Charging Input requirements

The amount and type of the input charging data differs significantly from one charging model to another. In function of the selected model and how the operator wants to implement it, the following input requirements can be reduced. The following type of information can be collected, when applicable, by each Serving Element and sent to the Charging Control. This requirement applies to both on-line and offline interfaces:

- *Subscription identity*. To identify the subscription who used the service.
- *Home environment identity*. To identify the home environment (e.g. service provider) of the user.
- *Local network identity*. To identify the network serving/providing access to the user.
- *Access technology* used during the provision of a service (e.g. GSM, UMTS, ISDN Access). Different access types have different capabilities and affect the services that can be provided to the user.
- *Destination endpoint identifier* for service requested (e.g. called number). To identify the destination of the connection.
- *Session Identifier (or unique charging ID).* To correlate chargeable info from several Serving Elements, but related to the same session/service.
- *Requested QoS*. QoS are usually derived from other parameters, such as media type (audio, video), connection type (conversational, streaming), CODEC type, bandwidth, maximum delay, jitter etc.
- *Negotiated QoS*. The above mentioned parameters affecting the requested QoS may change due to other end point having different capabilities, or the network limiting the usage of some resources. The negotiated QoS is the set of parameters that have been negotiated between the end points, taking into account the preferences in the network.

- *Allocated QoS*. The network might not be able to provide exactly the QoS that was requested/negotiated e.g. due to congestion. Therefore the QoS that was actually provided to the user should be available as charging input.

- *Update in QoS*. Every change in the parameters affecting the QoS should be registered. This applies to Requested QoS and Negotiated QoS (i.e. the user modifies the connection).

- *Resources allocated to the user.* To identify what network resources were eventually allocated to the user.

- *Time at which resources were provided for the service.*

- *Time at which resources were modified for the service*

- *Quantity of data transferred both to and from the user*. To identify the data volume that can be used as input for volume based charging.

- *Number of events*. If the message/CDR contains charging information for more than one similar event (e.g. number of received SMSs for the same user), the number of events should be included in the report.

- *Cause of service termination*. If the service terminated abnormally, the detailed reason should be provided.

- *Time at which service was terminated.*

- *Record Sequence Number*. Consecutive number for each CDR generated in the Serving Element in order to detect lost/duplicated CDRs.

4.9 Security requirements

4.9.1 Authentication

The Serving Element must authenticate Charging Control before setting up a session, according to a pre-determined security policy set by Charging Control. The security policy may vary between different applications. Authentication will usually be made either based on certificates or pre-shared secrets.

All requests from Serving Elements shall be accepted by Charging Control during a session. The maximum time a session is allowed to live must be limited. Re-authentication is needed after the session is terminated. The maximum session time must be configurable.

4.9.2 Transport Security

To provide sufficient security of data transmission to and from Charging Control the Serving Element must guarantee:

- Integrity (i.e. it must not be possible to change anything in the original information) of all information sent to Charging Control.

- Confidentiality (i.e. it must not be possible to read the original information) of all information sent to Charging Control. This is especially important if open interfaces (Internet), and open protocols are used from untrusted domains.

The requirement to use the Diameter protocol implies that there are specific requirements on encryption to secure Diameter messages, i.e. the Diameter protocol must not be used without any security mechanism (TLS or IPsec). It is suggested that IPsec can be used primarily for intra-domain traffic. It is also suggested that inter-domain traffic would primarily use TLS.

4.9.3 Border protection

All IP-interfaces between Serving Elements and Charging Control must be protected by means of border protection (e.g. firewalls and proxies) to prevent the possibilities of external attacks and intrusion (e.g. DoS attacks) on Charging Control.

4.9.4 Key management

Serving Elements data exchange with Charging Control must support key management as proposed by Charging Control, i.e. at least manual key management. The key management depends on protocols used and which type of authentication is used and may vary between different applications. This is further detailed in the security policy for the application.

4.10 Data Records

The Call Detail Records (CDR) [42] and Internet Protocol Detail Records (IPDR) [43] (http://www.ipdr.org/) are two data structures for accounting records. In addition, RADIUS Accounting Records (RAC) and the DIAMETER attributes are important. As said before, the RFC (Request for Comments) 2924 [39] summarizes existing IETF and ITU-T labor and talks about advantages and drawbacks on accounting attributes and record formats in more detail.

CDRs are sometimes named Call Detail Reporting or Call Data Records. They originally come form telephony based telecommunication systems and are well-known records for call-specific data. CDRs define the information to be shared between different domains and to be collected in its own domain. A CDR contains a complete data record about each session established, i.e. dialed digits, the calling party's phone number, call direction, service type, associated inverse multiplexing session and port, date, time, off-hook time, on-hook time, and a circuit identifier.

All telephony switches, Private Branch Exchanges (PBX), and ATM (Asynchronous Transfer Mode) switches produce CDRs. However, each switch product produces CDRs in different formats. Therefore, software needs to convert various CDR formats into a standard format usable by a charging system. More information about CDR can be found in the document "Call Detail Records for UNI 1.0 Billing" [44].

The IPDR.org is an open consortium of service providers, equipment vendors, system integrators, and billing and mediation vendors collaborating to facilitate the exchange of usage and control data between network and hosting elements and operations and business support systems by deployment of Internet Protocol Detail Record (IPDR) standards. It refers (1) to a functional operation, where a Network Data Management (NDM) function collects data from devices and services in a provider's network, and (2) to usage, the type of data, which shows an open, extensible, and flexible record format (the IPDR record) for exchanging usage information of essential parameters of IP-related transactions.

Additional data formats are available, however, mainly with respect to a particular protocol as mentioned above or an application. DNS and DHCP maintain subscription profile data, which form a type of standardized data format, and LDAP offers mechanisms with transfer capabilities for subscription profile data. However, these data formats are not generally exploited for the purpose of accounting or other AAA tasks.

5 SIP

5.1 General overview of SIP

The Session Initiation Protocol (SIP) [6] is an application layer control protocol used for establishing, modifying, and terminating multimedia sessions between users. SIP provides the necessary signaling for initiating communications and it supports user and device mobility by means of SIP servers. These servers can operate in either redirect or proxy mode. SIP is independent of the lower layer transport protocol, it can either use TCP, SCTP (Stream Control Transmission Protocol), or UDP as a transport protocol. The default port for SIP depends on the transport protocol in use. It is 5060 for UDP, TCP, and SCTP; and 5061 for Transport Layer Security over TCP (TLS). SIP also makes use of the Session Description Protocol (SDP) [7] (see next chapter). By carrying SDP messages inside an INVITE payload, to describe the media content of the session.

The protocol is text-based and similar to HTTP by reusing its message structures and error codes. SIP does not define a Type of Session by default, so it is usable for any kind of service needing management of sessions, like video/audio conferencing, interactive games, and instant messaging.

Users need to register theirs IP address at the SIP registrar responsible for their domain to identify the actual location of the user in terms of an IP address. Thus, when inviting a user, the caller sends his invitation to the SIP proxy responsible for the user’s domain, which checks in the registrar’s database the location of the user and forwards the INVITE to the callee. The callee can either accept or reject the invitation. The session setup end when the calling party receives the acknowledgment. During this message exchange, the caller and callee exchange their contact address that is the address at which they would like to receive the media. When the session setup finishes, the end systems can transfer data, through a lower OSI (Open System Interconnection) layer, directly without involving the SIP proxies.

The EuQoS project has decided to use digest authentication mechanisms for authenticating a user. However, SIP can also use other authentication protocols such as the Secure Real-time Transport Protocol (SRTP) [64] and the Multimedia Internet KEYing (MIKEY) [65] protocol. The digest authentication mechanisms are based on a challenge/reply approach. In this document it will be assumed that a roaming user is supposed to contact a local SIP proxy in the foreign network (the user’s local network).

The SIP registrars provide user localization services. In combination with the session management capabilities, it provides several levels of mobility; in particular user and session mobility. However, SIP itself does not provide the user's network location. If this information is required it should be done via SIP extensions such as SDP, which is used during the setup of sessions.

If any of the required capabilities can not be provided by the user agent, some re-negotiation has to take place. This could, for instance, establish a reduced session, i.e. simple audio instead of audio **and** video communication.

SIP does not provide any control of the subsequent data flow between endpoints, nor does it provide any resource reservation mechanism, because SIP messages are carried independently of the subsequent session content. To solve this, EuQoS introduces an enhancement of the SIP protocol that allows resource reservation (see chapter 6). SIP only contacts the peer with whom the user would like to establish a session and relies on other protocols, such as SDP, for any subsequent user data exchange. During a session, SIP allows users to modify the session's communication parameters and to keep track of all on going sessions.

Finally, two important ideas to keep in mind are that SIP does **not** provide services, but rather provides primitives that can be used to implement these services; and that SIP works with either IPv4 or IPv6. SIP makes use of an *offered/answerer* model, in which the caller represents the *offered* and the called party represents the *answerer.*

5.2 Logical entities in SIP

A usual SIP network contains six basic components: calling and callee User Agents (UA), proxy, redirect, registrar, and location servers. This is a logical division, but these components may execute within the same physical entity.

5.2.1 SIP user agents

A *SIP User Agent* is a logical entity that can act as both a user agent client (UAC) and user agent server (UAS). The User Agent Client initiates a request and the User Agent Server generates a response (to accept, redirect, or reject a request).The client is able to send invitations for a session to a peer and acts as a client for the duration of the session. If it receives a request it assumes the server role. The UAs settle the session parameters.

5.2.2 SIP servers

A *SIP Server* is an entity that receives SIP requests. It is able to process these requests and to send replies. A SIP Server can operate in a redirect or in a proxy mode:

Redirect Server

A SIP redirect server generates a response indicating the address where the requester can contact the next proxy server or UA.

Proxy Server

A SIP proxy either serves a request or forwards this request to another server, which is more capable of serving the request. A proxy server is able to modify messages and acts as a client on behalf of the requesting user. Thus, it contacts external location servers to determine the target user's location. In addition, there are three different types of proxy servers [55]:

- *Stateless proxy server:* does not keep a record of transactions, thus stateless proxy servers behave as simple message forwarders.

- *Stateful proxy server or transaction stateful proxy:* these proxies keep a record of state during the transactions, which saves information about all requests they receive and send within a session. This information is then used to process future messages associated with that request.

- *Call stateful proxy:* stores all the state pertaining to a session (e.g., from INVITE to BYE). A call stateful proxy is always a transaction stateful proxy, but not the other way round.

Registrar Server

A registrar server receives registration messages (i.e., REGISTER requests) from user agents, extracts information about their location, and stores that information in a database (to implement a Location Service).

A *Registrar* is a server that receives REGISTER requests and processes those belonging to the domain it handles, by updating a location database based on the information carried by the message.

Location Service

The Location Service (or Location Server) stores information about the location of the users, and provides that information to proxy and redirect servers when requested. SIP entities can use the Domain Name System (DNS) to locate SIP servers (See section 4.7)

An example of the basic architecture of a SIP network (the SIP Trapezoid) is shown in the following figure.

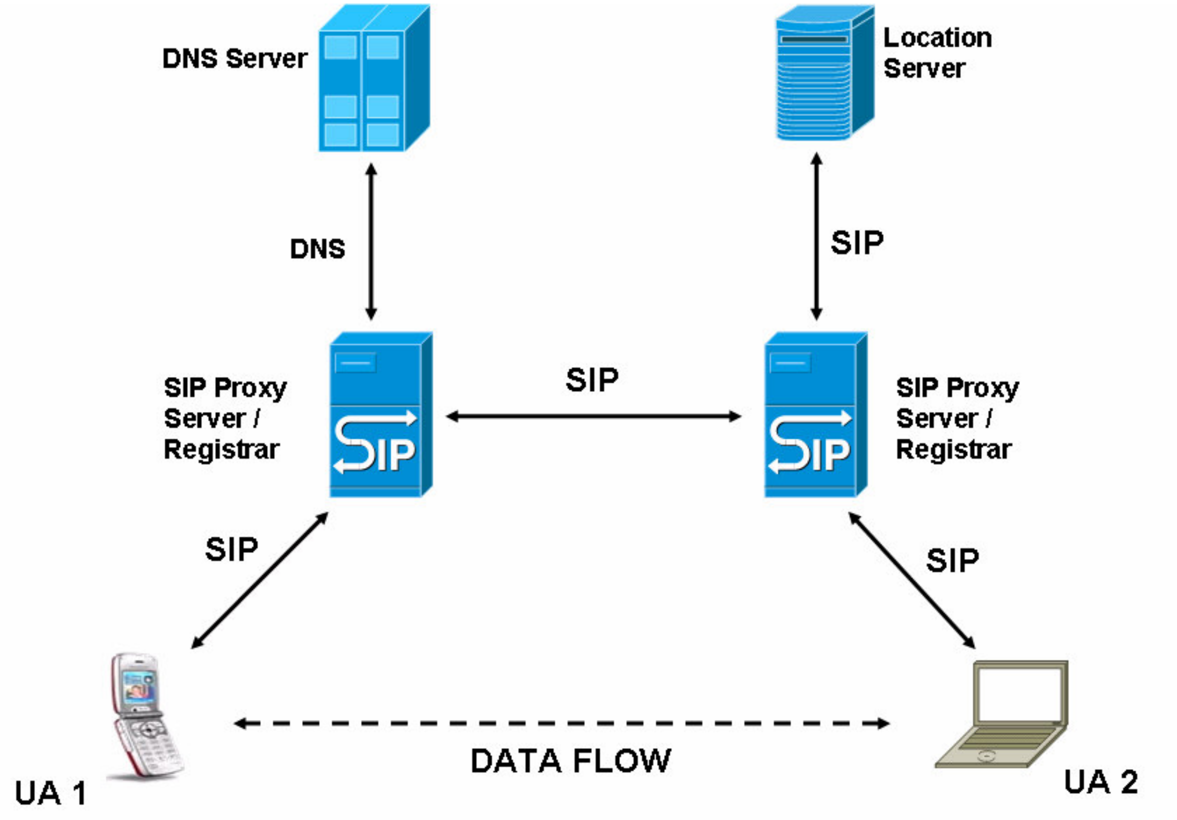


Figure 5: SIP architecture

5.3 SIP addresses

A SIP user is identified through a SIP Uniform Resource Identifier (URI) [56] in the form user@domain, where domain can also be an IP address. This address can be resolved to a SIP proxy, which is responsible for the user's domain. In addition, a SIP URI can identify a user, a specific device, or an instance of a user at a given UA.

At least, SIP URIs and SIPS URIs are supported, although others (such as TEL URL) are commonly supported.

- sip:Jesus.Miguel.Gutierrez-Barquin@ericsson.com
- sips:Jesus.Miguel.Gutierrez-Barquin@ericsson.com
- tel:+34669094123
- sip:proxy.kth.com:5060
- sip:another-proxy.kth.com;transport=UDP

It is a requirement that the SIP and SIPS URIs include a host name. However, they may also include username, port numbers, and/or other parameters. The address space is unlimited, and non SIP/TEL URIs are also valid under certain contexts as HTTP, IM, PRES, and MAILTO.

5.4 SIP messages

The SIP type of message is carried in the request and identifies the action that the requester wants to invoke at the server. There are specific fields which are mandatory in the header depending on the method.

- **INVITE**: it is used to establish a session between peers or servers. As an example, and INVITE request is shown below:

  ```
  INVITE sip:jesusm@kth.se SIP/2.0
  Via: SIP/2.0/UDP pc3.kth.se;branch=z9hG4bK776asdhds
  Max−Forwards: 70
  To: Bob <sip:jesusm@kth.se>
  From: Alice <sip:alice@kth.se>;tag=1618303774
  Call−ID: s424b4c76e32450@pc3.kth.se
  CSeq: 232159 INVITE
  Contact: <sip:alice@pc3.kth.se>
  Content−Type: application/sdp
  Content−Length: 142
  ```

  The first line identifies the method name: INVITE in this case. The following lines are the minimum required fields of this type of request.

  - **Via** header field contains the path followed by the request to reach the recipient, and the reverse of this path should be followed by responses. Each node that the INVITE request goes through adds a new *Via* field.

  - **Max−Forwards** limits the number of hops to the destination. When the count down gets to zero, the request will not be longer transmitted.

  - **To** indicates the display name and the SIP URI of the logical recipient of the request. The *To* header field may contain an IP address, which can identify a user or a resource.

  - **From** identifies the originator of the request. This field contains the SIP URI and it might also include a display name. The *From* header field must contain a tag parameter; this is appended to the field value and identifies a dialog.

  - **Call–ID** is a globally unique identifier for an invitation. It is always copied in the response to a particular request and it must be a globally unique identifier.

  - **CSeq** stands for Command Sequence. It is an integer used as a traditional sequence number, distinguishing new requests from request retransmissions.

  - **Contact** is the SIP URI where the user agent would like to be reached. If it were a REGISTER request, the Contact URI would be associated with a preference value among the given locations, named "q".

  - **Content–Type** describes the media-type of the message.

  - **Content–Length** is an integer that defines the length of the message body in octets. If no message is included, the field value must be zero.

There are some other header fields that may appear in the request such as **Accept, Allow, Content-Disposition, Content-Encoding, Expires, In-Reply-To, Priority, Require, Retry After, and Supported.** Section 20 in SIP standard [6] describes these fields in a deeper way. The details of the session to be established are not explicitly described by SIP. They are usually carried in the SIP message body using the *Session Description Protocol* (SDP) [7] (see chapter 5).

- **ACK**: it is the response to an INVITE request, i.e. acknowledge. An ACK confirms that the client has received a final response to an INVITE request.

- **CANCEL**: is used to cancel a previous INVITE request sent by the client.

- **BYE**: it is used to terminate a session. The BYE message terminates the communication session without requiring any acknowledgment. This message may be sent by either the originator of the call or by the receiver.

- **OPTION**: allows a user agent to know about the capabilities to process messages. It may be used for the negotiation of the supported methods and extensions. SIP allows the definition of new methods to facilitate extensibility. These definitions don't need to be supported by all SIP user agents; thus, UAS can use the OPTION method to check if a given UA supports the desired extension.

- **REGISTER**: this method is used to register a device address within a system via a Registrar Server. The registration process binds a SIP URI address to the device's current location. A user agent is able to update and modify bindings stored in the location service by sending a new REGISTER message with a new contact address indicating the new location of the user. It is necessary for a device to perform the registration in order to provide location information to permit incoming calls.

5.4.1 Response messages

The SIP response codes are extended from the HTTP response codes. They consist of a three digit numeric status code. The first digit informs about the class that the code belongs to, and the other two digits define the response message [58]:

- **1xx Responses:** Informational Responses (e.g. 180 Ringing and 100 Trying).
- **2xx Responses:** Successful Responses (e.g. 200 OK).
- **3xx Responses:** Redirection Responses (e.g. 302 Moved Temporarily).
- **4xx Responses:** Request Failure Responses (e.g. 404 Not Found).
- **5xx Responses:** Server Failure Responses (e.g. 503 Service Unavailable).
- **6xx Responses:** Global Failure Responses (e.g. 600 Busy Everywhere).

Event notification

The Event notification extension [59] provides additional capabilities to SIP entities, which can request notification of a particular event. An event package is a document that specifies the event extension and defines the syntax and the semantics of the event information.

This extension defines SUBSCRIBE and NOTIFY methods, in order to ask for notification and to obtain notification when the state change occurs respectively.

SUBSCRIBE

The SUBSCRIBE method is used to request notification of events and to establish a subscription. A subscription is always associated with a dialog, that is specified by the dialog ID, to an event package and it can be identified with an "id" parameter if there are multiple subscription in the same dialog. The duration of the subscription is indicated in the value of the Expires header or by a default value.

NOTIFY

A NOTIFY request contains the event package name, an identification of the subscription, and an optional body, which can contain the state of the subscribed resource.

5.5 SIP Registration

A user has to register its SIP Universal Resource Identifier (URI) and current location in order to be reachable via this URI. SIP defines a particular type of SIP Server, the registrar server, which is able to process incoming REGISTER requests and to update a location service database. The binding between a SIP URI, known as address-of-record (AOR), and contact addresses is stored in the location server responsible for that particular domain. A REGISTER request is able to add or modify the bindings.

Therefore, a SIP user is reachable by looking at the location service database, and the SIP server redirects or proxies the requests to the new location. The registration establishes a dialog between the SIP User Agent from the registration party and the SIP registrar server.

A SIP User Agent uses the REGISTER method in the SIP header, where the Contact header indicates the new location. The number of Contact headers could be more than one. In this case, the "q" parameter indicates the preference for the Contact header value. The Expires value informs of the time that the binding is valid.

During Registration UA send information about their current location to their Registrar Server by means of a REGISTER message. The picture below shows the messages involved in the registration process.

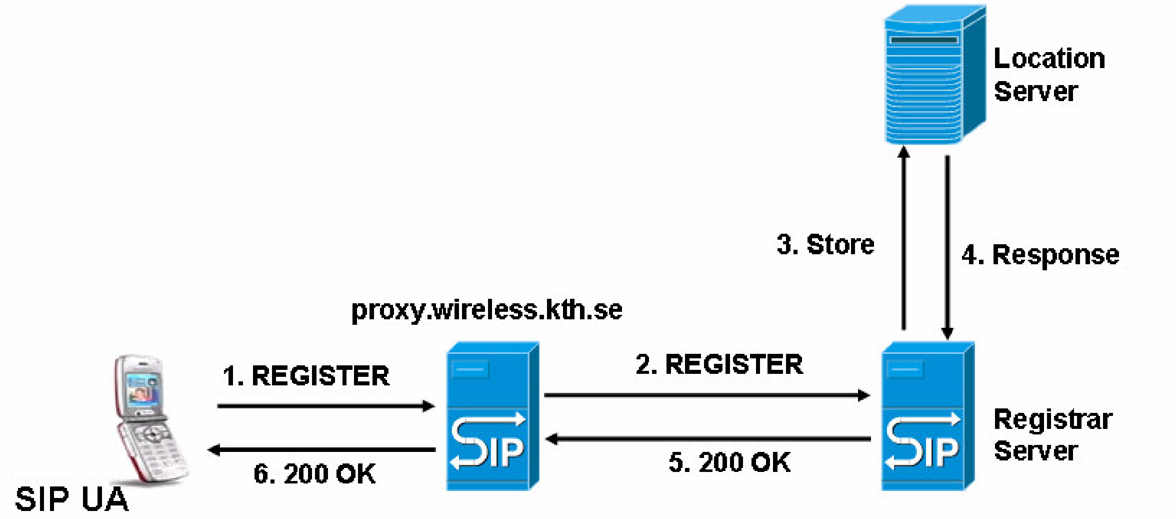


Figure 6: User registration

```
REGISTER sip:wireless.kth.se SIP/2.0
Via: SIP/2.0/UDP proxy.wireless.kth.se:5060
From: sip:jesusmiguel@wireless.kth.se; tag=f32428b4350c95f7e
To: sip:jesusmiguel@wireless.kth.se
```

```
Call-ID: 52s2fdd7ac49d1fg@wireless.kth.se
CSeq: 2 REGISTER
Date: Sun, 04 Dec 2005 17:14:23 GMT
Contact: <sip:jesusmiguel@wireless.kth.se:5060>; q=0.8
Contact: <mailto:jesusm@kth.se>
Contact: <sip:jesusmiguel@ericsson.home.com>; expires=3600
Expires: 7200
Content-Length: 0
```

The UA sends a REGISTER request. The kth server forwards the request to the Registrar Server and the registrar inspects the URI to determine if it is responsible for the AOR of the specified domain. If so, the registrar processes the message and stores the UA's current location in the Location Server. When the registration succeeds, the registrar returns a "200 OK" response message which contains all current bindings and their status.

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP proxy.wireless.kth.se:5060
From: sip:jesusmiguel@wireless.kth.se; tag=f32428b4350c95f7e
To: sip:jesusmiguel@wireless.kth.se; tag=f32428b4350c95f7e
Call-ID: 52s2fdd7ac49d1fg@wireless.kth.se
CSeq: 2 REGISTER
Date: Sun, 04 Dec 2005 17:14:24 GMT
Contact: <sip:jesusmiguel@wireless.kth.se:5060>; q=0.8
Contact: <mailto:jesusm@kth.se>
Contact: <sip:jesusmiguel@ericsson.home.com>; expires=3600
Content-Length: 0
```

5.5.1 Update registration

Bindings expire if they are not refreshed. A user agent can update bindings and also include new Contact addresses or modify their location by sending a REGISTER request.

- **Delete locations:** A UA can set the expiration timer of a location to "0" for a certain contact address to erase this location from the server. Instead, the UA may want to delete all bindings. In such case, the UA needs to set the Contact header to "*" and the Expires value to "0".

- **Refresh locations:** When a user agent wants to refresh bindings, it sends a REGISTER request for each binding. If bindings are not periodically refreshed, they expire.

5.5.2 Discovery of a registrar

The simplest way to contact the registrar is to manually configure the user agent with the SIP registrar's address. Although this is suitable for a static scenario, it is probably not acceptable in a dynamic environment. EuQoS utilizes only one SIP register server per domain, so that, the EuQoS system uses a static configuration.

5.6 Session setup

This subsection presents an example of the use of SIP between two end users. This example is related to the *SIP Trapezoid* and it only shows a simple SIP message exchange. **Figure 7** illustrates the basic call message flows. Two users want to establish a media session, while they are located in different domains.

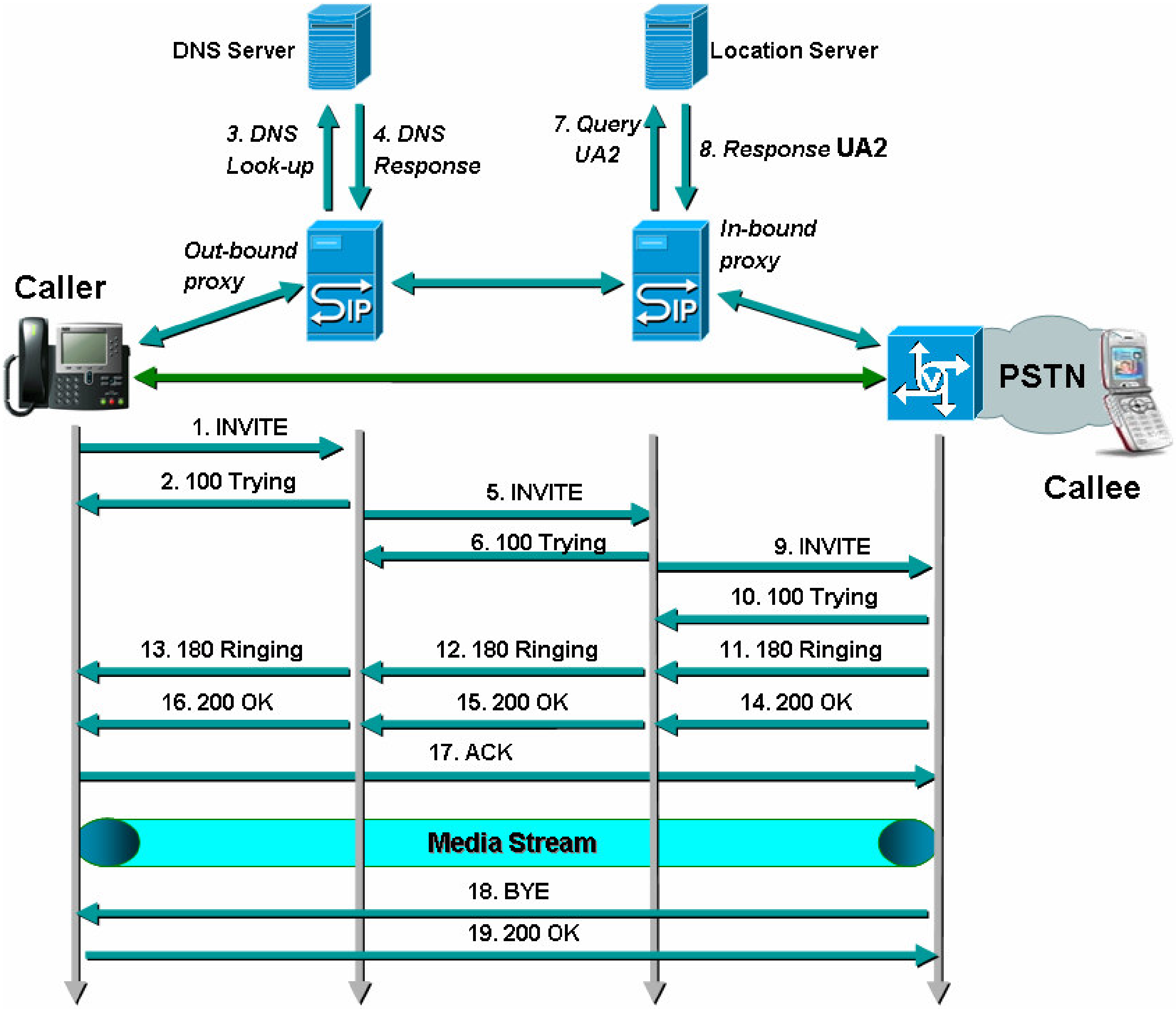


Figure 7: Call set-up

Description of the messages involved in the call set-up:

1. INVITE

This message is sent by the caller (SIP UA1) to invite the callee (SIP UA2) to start a session. The UA1 sends the INVITE message to the Out-bound Proxy. The details of the session (i.e., ports, type, supported CODECs, and media protocol) are defined in the SIP message body within a SDP datagram.

Messages 2, 6, and 10 "100 Trying"

The SIP Server sends back a "100 Trying" message when it receives the INVITE request. This indicates the correct reception of the INVITE message.

3. DNS Look-up

This is a query to the DNS server, used to resolve the callee's address. This is a non-SIP message.

4. DNS Response

The DNS server sends this message as response to the DNS look-up. This message contains the IP address and port of the callee's domain In-bound Server. This is also a non-SIP message.

5. INVITE

The Out-bound proxy forwards the INVITE towards the In-bound Proxy associated with the domain of the URI. This INVITE may need to go through several domains or SIP proxies before reaching the In-bound Proxy of the called party. Note that this has been simplified in the example, where the Out-bound Proxy communicates directly with the In-bound Proxy.

7. Query UA2

The In-bound Proxy queries the current location of the UA2 (non-SIP messages).

8. Response UA2

Response sent by the Location Server to inform about the UA2 location. The UA2 must register in advance in order to be reachable at this point.

9. INVITE

The Inbound Proxy forwards the INVITE message to the SIP UA2.

Messages 11, 12, and 13 "180 Ringing"

The called UA starts ringing and sends this message through the network to the caller.

Messages 14, 15, and 16 "200 OK"

When the user of SIP UA2 accepts the call, a "200 OK" message is sent through the network to the caller.

17. ACK

The caller confirms to the called party the reception of the "200 OK" SIP message.

Once a session is established, peers can modify the existing session, by sending a new INVITE request, known as RE-INVITE, within the same dialog. The media data exchange can start from now on. This data is routed directly between end-points. In other words, data in a media session does **not** need to follow the same path as the SIP signalling did.

18. BYE

The session terminates by sending a BYE message for a particular session.

19. 200 OK

Confirms the reception of the BYE message. The communications ends in this point.

5.7 SIP Servers location

For correct session establishment, the calling party needs to forward the call to the right domain. When a user agent sends a request to the proxy server, this proxy will forward the request to the next proxy server or to a user agent.

A SIP entity performs a DNS query whenever it contacts another entity located by SIP URI. In other words, once the request reaches the *out-bound proxy* (the proxy server in charge of sending SIP messages to other domains), the SIP client in this server uses DNS procedures (i.e., DNS SRV [60]) to resolve the destination SIP URIs into its IP addresses. The DNS SRV record lists the available SIP servers, and then the current server forwards the request to the most appropriate server. This request can go through several intermediate servers before it arrives at the called party. Additionally, with suitable DSN SRV records, DNS allows servers to send responses to a back-up client when the primary client fails

A similar process is performed at all intermediate SIP servers until the request reaches the “in-bound proxy” of the called party domain (see next figure).

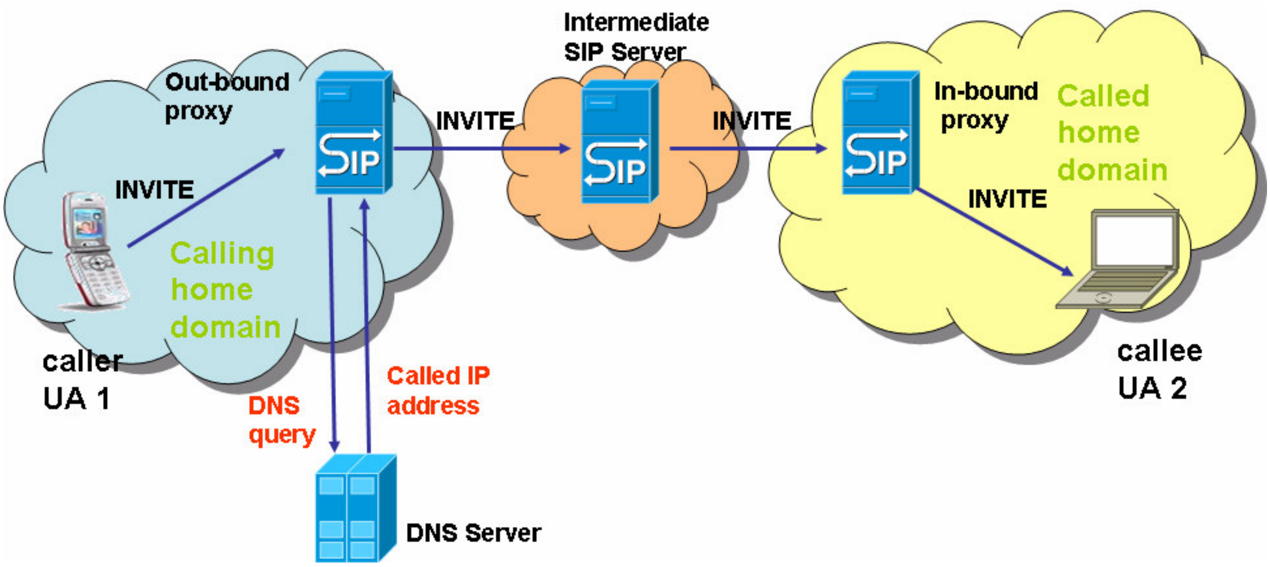


Figure 8: Locating SIP Servers

Another method to locate a SIP proxy is based on the use of DHCP. More information about this procedure can be found in the standard [61].

6 The Session Description Protocol

SDP is a session description protocol for multimedia sessions [7]. SDP is used to describe the set of media streams, CODECs, and other media related parameters supported by either party.

All SIP implementations must support SDP, although they can support other bodies. The session description protocol was initially developed to support multicast sessions in the Internet. However, it has been gradually adjusted to SIP purposes. SDP is also used by other protocols than SIP, such as RTSP and SAP.

SDP includes:

- The type of media (video, audio, etc)
- The transport protocol (RTP/UDP/IP, H.320, etc)
- The format of the media (H.261 video, MPEG video, etc)

When SDP is conveyed by SIP, many SDP session descriptions may be concatenated together (the `v=' line indicating the start of a session description terminates the previous description). Some lines in each description are required and some are optional but all must appear in exactly the order given here (the fixed order greatly enhances error detection and allows for a simple parser). Optional items are marked with a `*'.

Session description

- v= protocol version
- o= owner/creator and session identifier
- s= session name
- i=* session information
- u=* URI of description
- e=* email address
- p=* phone number
- c=* connection information - not required if included in all media
- b=* bandwidth information
- z=* time zone adjustments
- k=* encryption key
- a=* Zero or more session attribute lines (media descriptions)

Time description

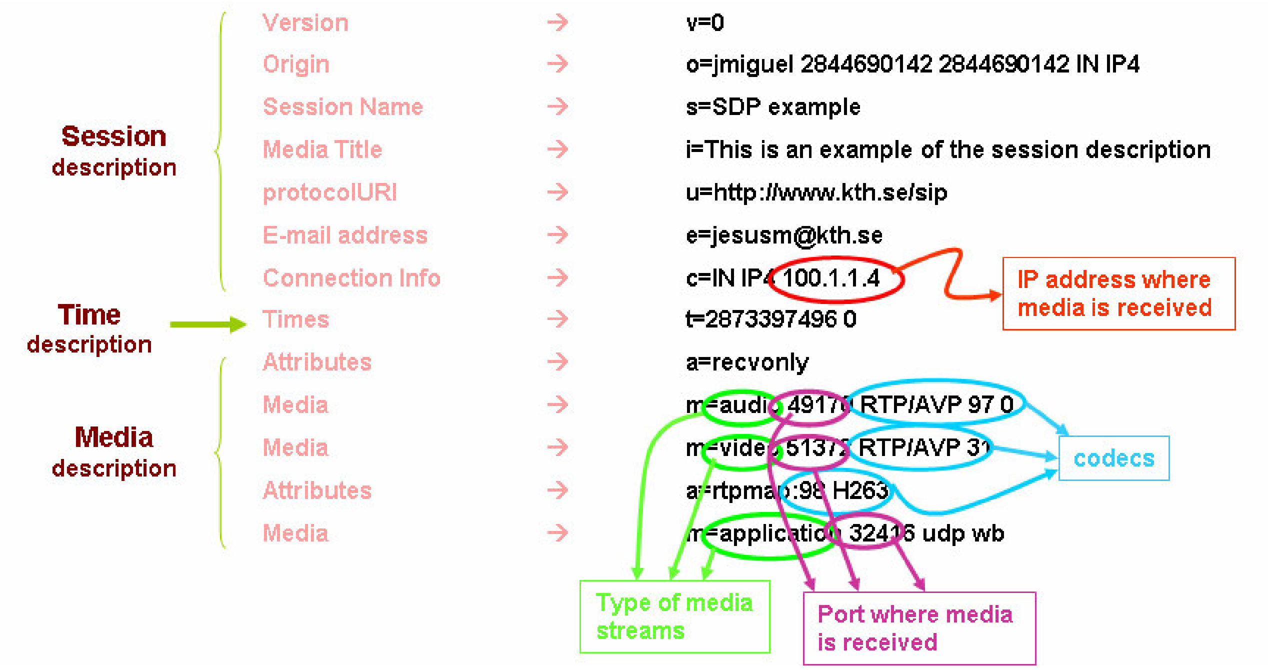
- t= time the session is active
- r=* zero or more repeat times

Media description

- m= media name and transport address
- i=* media title
- c=* connection information - optional if included at session-level

- b=* bandwidth information
- k=* encryption key
- a=* zero or more media attribute lines

An example SDP description is:



```
v=0
o=jmiguel 2844690142 2844690142 IN IP4
s=SDP example
i=This is an example of the session description
u=http://www.kth.se/sip
e=jesusm@kth.se
c=IN IP4 100.1.1.4
t=2873397496 0
a=recvonly
m=audio 49170 RTP/AVP 97 0
m=video 51372 RTP/AVP 31
a=rtpmap:98 H263
m=application 32416 udp wb
```

Figure 9: Session Description Protocol example

More information about this protocol can be found in [7] and [19].

7 The EuQoS end-to-end architecture

The EuQoS end-to-end architecture has two views; a network deployment view across a number of Autonomous Systems (AS) domains, and software view, overlaid on the network view, within an AS.

A first approach to the signalling in the EuQoS architecture can be seen in Figure 10, where A-SSN stands for Application Signalling and Service Negotiation.

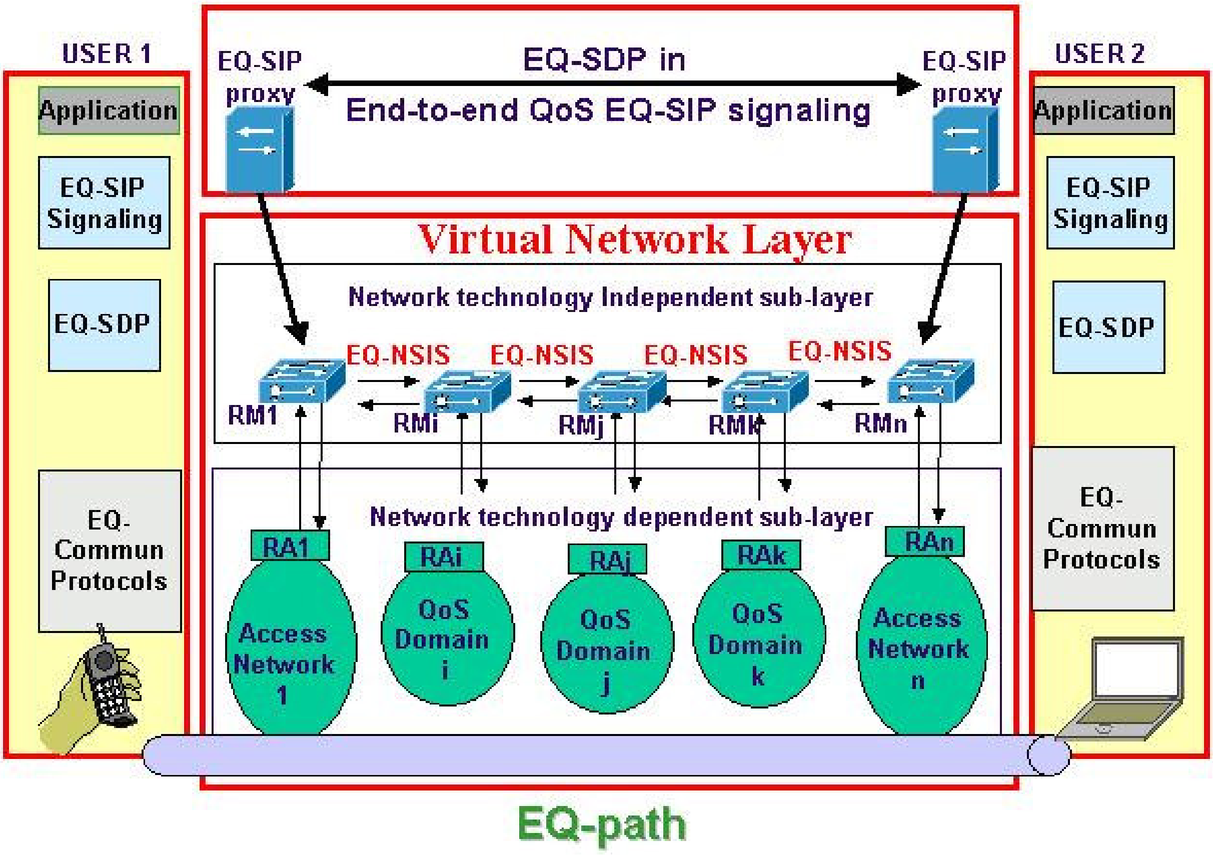


Figure 10: Signalling in the EuQoS architecture

From the horizontal view (i.e. as different planes), this implies a clear separation between the application signalling level and the resource management level. The Control Plane has interactions with both a technology independent layer, based on a Resource Manager (RM), and a technology dependent layer, based on a Resource Allocator (RA). From the vertical view (i.e. different network partitions), this "divide and conquer" approach implies a clear separation between the various access technologies and the different core networks involved in the end-to-end connection.

The other ideas that govern the EuQoS solution occur because of the need for synchronisation between the Service and Control Planes; specifically, the Control Plane is solicited when an application needs some network resources. The aim of the EuQoS architecture is to provide QoS only for those applications, which need them, and only when they are needed. For this reason, the EuQoS system is based on the session concept.

First, an application sets up a session, which triggers the corresponding network QoS setup. This synchronizes the QoS requirement / setup with the usage of QoS by the application. Furthermore, trying the resource allocations to the session also ensures the graceful release of QoS resources when the application terminates. For this purpose, the EuQoS system uses an enhanced version of the Session Initiation Protocol (SIP) [6], named EQ-SIP (described in section 2.4), which allows QoS negotiation during the session establishment.

QoS preconditions

Current standards, specifically RFC 3312 [20] (Integration of Resource Management and Session Initiation Protocol (SIP)) and RFC 4032 [21] (Update to the SIP Preconditions Framework) introduce the concept of QoS precondition, i.e. a set of constraints about the session to be established, thus the session cannot be established until these QoS precondition are fulfilled.

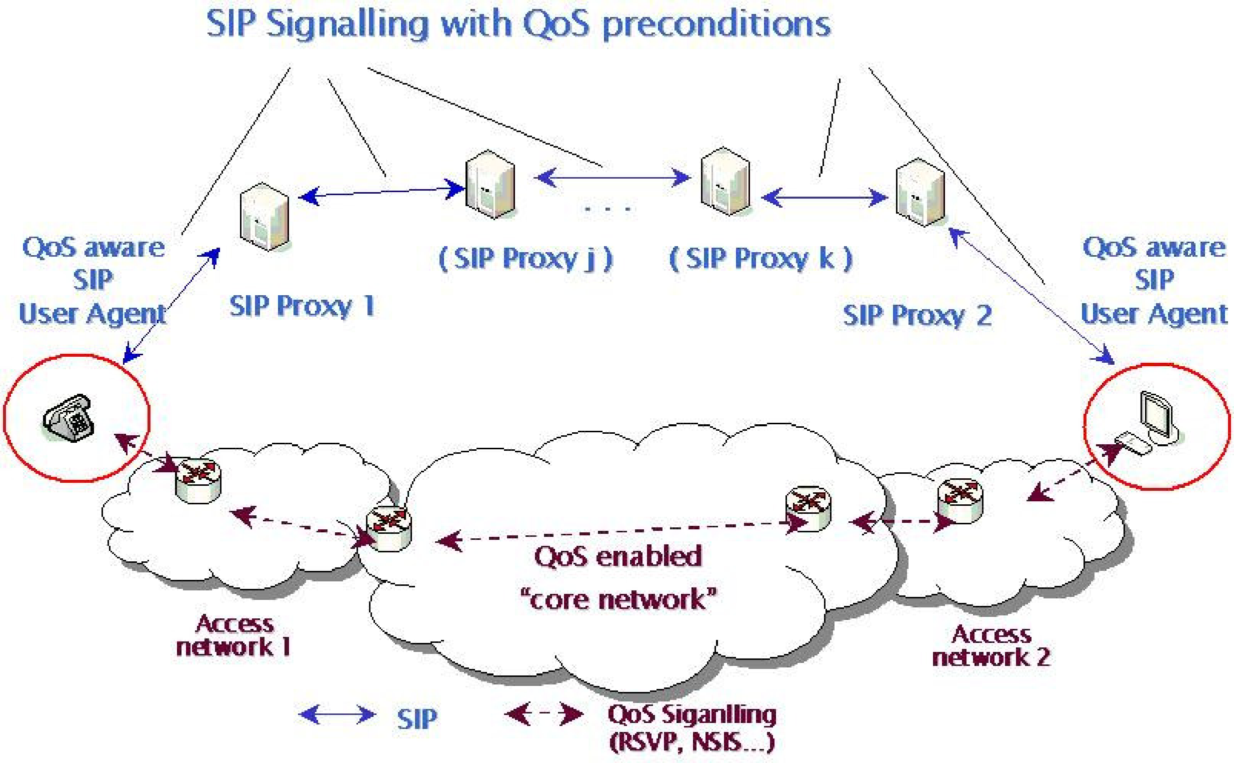


Figure 11: SIP & QoS: current status

The objective of these preconditions is to ensure that resources are made available *before* the called device rings. The setup of the QoS reservations is a process originated by the calling user agent. This model is not restricted to Resource Reservation Protocol (RSVP) [17], but it was originally designed with RSVP in mind.

The resource reservation handled by the user agent increases complexity of the terminal itself, which is a potential problem. This problem could be critical for light terminals (small IP devices or other handheld IP based terminals), because if the intelligence resides in the net, the terminals are simpler than if the intelligence is in the user agent.

The QoS preconditions are media stream specific. They are specified in Session Description Protocol (SDP) [7] messages as attributes/parameters of the media. The following media attributes are basic in the request for QoS:

- **Current status**: The current status attribute carries information about the current status of network resources for a particular media stream.

- **Desired status**: The desired status attribute carries the preconditions for a particular media stream. When the direction-tag of the current status attribute, with a given precondition-type/status-type for a particular stream is equal to (or better than) the direction-tag of the desired status attribute with the same precondition-type/status- type, for that stream, then the preconditions are considered to be met for that stream.

- **Confirmation status**: The confirmation status attribute carries threshold conditions for a media stream. When the status of network resources meet these conditions, the peer user agent will send an update session description containing an updated current status attribute for this particular media stream. This is used to request the confirmation for resource reservations to the peer user agent.

The QoS preconditions are included in the SDP description rather than in the SIP header because these preconditions are stream specific.

Information carried in the status attribute:

- *Types of status*: end-to-end and segmented. The **end-to-end status** reflects the current or desired status of the end-to-end reservation of resources. The **segmented status** (*local* and *remote*) reflects the current or desired status of the access network reservations of both user agents involved in the session. End-to-end status is useful when end-to-end resource reservation mechanisms are available. The segmented status is useful when one or both User Agents (UA) can perform resource reservations in their respective access networks, but there is not end-to-end resource reservation.

- *Precondition strength*: indicates whether or not the callee can be alerted, if the network **fails** to meet the preconditions.

- *Direction of reservation*: direction in which a particular attribute (current, desired, or confirmation status) is applicable to.

NOTE: The values "local" and "remote" represent the point of view of the entity generating the SDP description.

7.1 Call setup

The calling SIP user agent initiates the communication when it sends the INVITE request. The resource reservation takes place if the called party is reachable and able to support the media as requested by the caller. An UPDATE signal informs each SIP Proxy along the path if the resource reservation attempt ended successfully. A ringing tone can start after this UPDATE signal arrives at the called party. Finally, the data stream begins, if the called party accepts the session. This process is clearly seen in the following diagram.

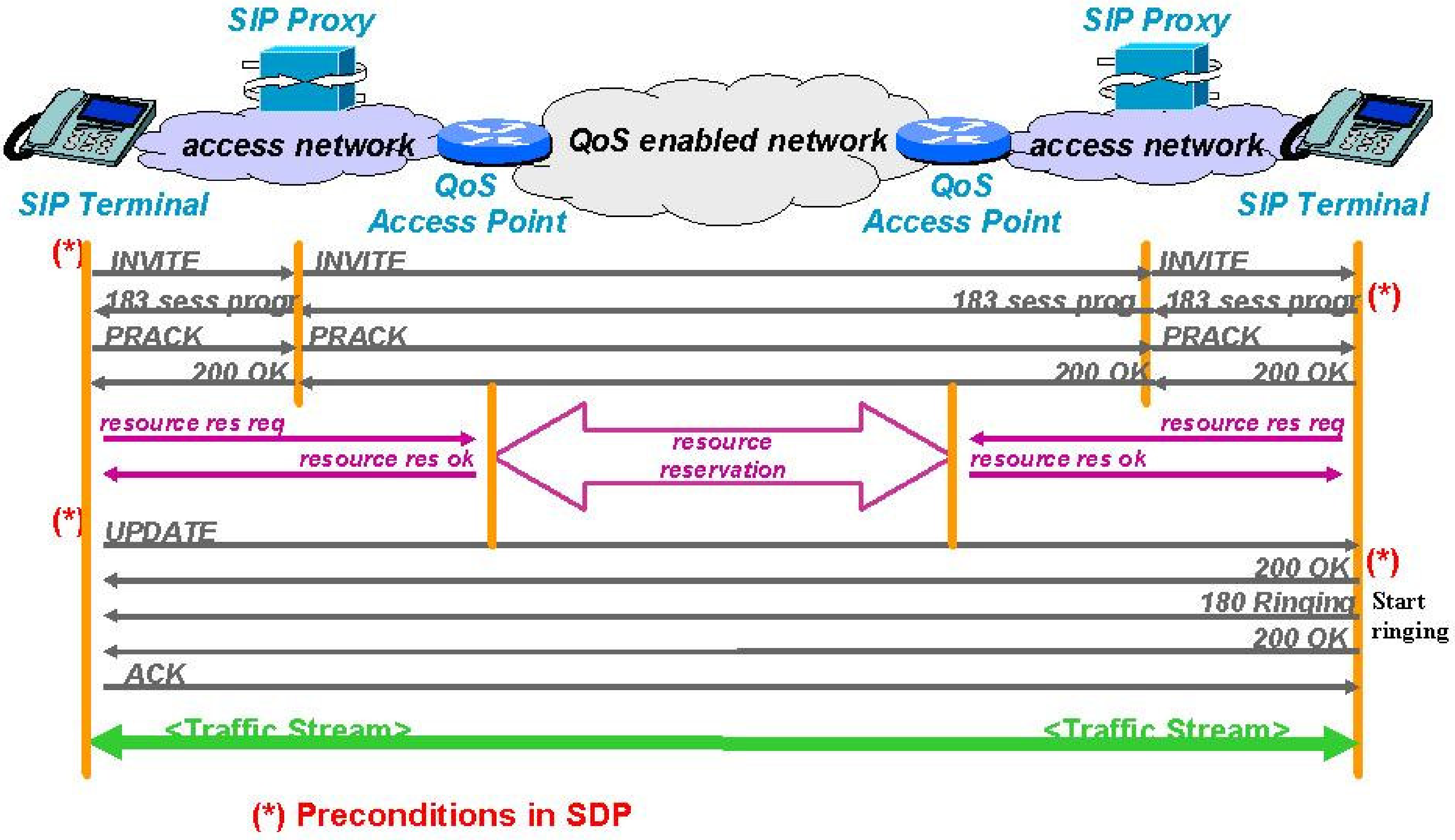


Figure 12: Call setup

Issue 1: The QoS is requested by user agents and related to SIP via preconditions (see section 7.3.1).

Issue 2: Preconditions only check that the QoS reservation has/has not been successful. It is not possible to negotiate QoS characteristics and agree on the QoS between the QoS aware entities in the SIP dialogue (see section 7.3.2).

In the current model the UAs must be "QoS aware" and capable of reserving the resources according to the "QoS language" spoken by the network. Details of how the user agents make these requests are addressed in [17].

1) The UAs are QoS aware and make QoS reservations:

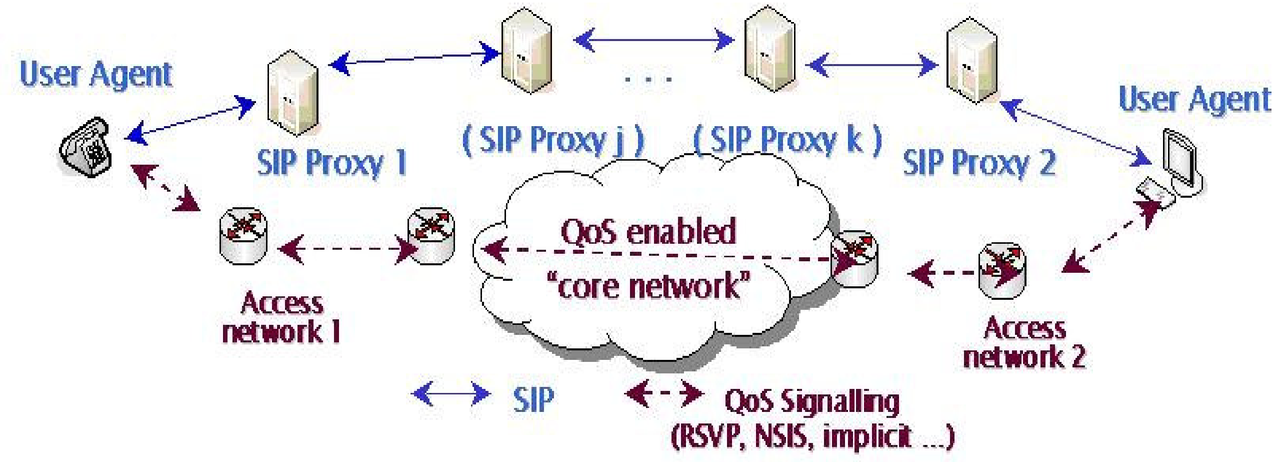


Figure 13: QoS aware user agents and QoS reservation

We can consider a second model where the UAs are QoS aware, but delegate to the SIP Proxies to make QoS reservations.

2) The UAs are QoS aware, but do not make QoS reservations directly:

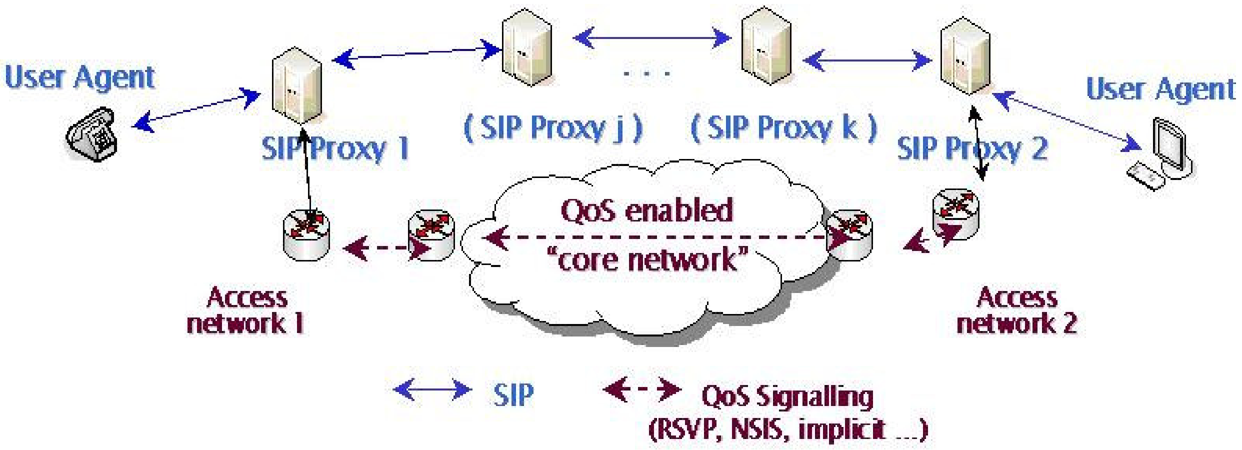


Figure 14: QoS aware user agents, but the SIP proxies make QoS reservation

Nevertheless, a third more general model should be supported, where the UAs need not be aware of QoS nor do they need to able to speak the QoS language of the network. This is shown in Figure 15.

3) The UAs are not QoS aware:

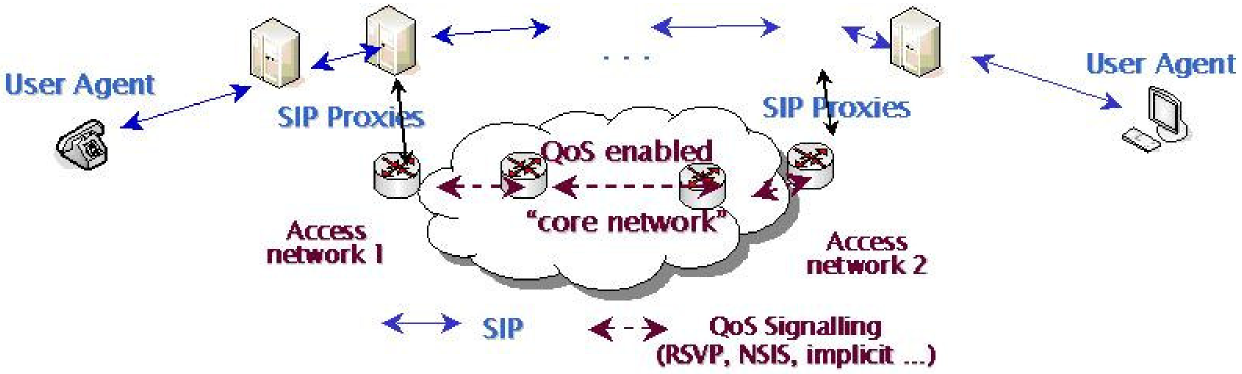


Figure 15: non-QoS aware user agents

Note that in the third scenario there are two functions that are performed by proxies on behalf of the terminal: (1) QoS negotiation and (2) resource reservation with the network. This least two possible implementations:

1. A “Gateway” performs the QoS negotiation and the proxy performs the resource reservation. This is very similar to scenario 2, with the “Gateway” playing the role of a user agent.

2. A single proxy takes care of both functions.

This third model has the advantage that user agents do not need to be upgraded and can be simpler. It also is in line with the current trend of having “Session Border Controllers” [62], to control SIP and media transmission at the network’s border.

Generally complexity in the network is both very expensive and scales poorly, whereas complexity in the handset scales with the number of handsets. Note that handsets are produced in very high volumes and undergo rapid development, while changes to the infrastructure are much harder and occur much more slowly. It is readily introduced without changes to then network and it is not tied to many operators introducing a feature, but is based on the user adoption via upgrade or new software installation in handsets.

However, in this case, the functionality is carried out in the network, i.e. the resource reservation has to be performed in the net. There is not any advantage if the handset is aware of the process since it mainly involves the network. This time is not possible to choose whether setting the intelligence in handsets or in the net since it is the net the one who has to fulfil the requested QoS.

7.2 QoS negotiation in SIP: the need for an enhancement

In the current QoS preconditions drafts, a user agent can only say it wants QoS and learn if QoS have been reserved. There is no way to express which level of QoS is desired per media stream or the target QoS by the end-user.

Unfortunately, the QoS the user actually wants cannot be derived accurately simply from information already available in the SDP; i.e. the CODEC and the **optional** bandwidth parameter (“b=”).

Additionally, end-users cannot reach a QoS agreement at session set-up using SIP alone as there is no way to ensure that in both access networks the end-user’s UA will each request and that both will be provided with the same QoS. Therefore it is impossible for service providers to deliver "predictable end-to-end QoS" to their customers, thus they cannot readily charge for a give QoS level of provisioning.

SIP proxies of the local access network or service provider cannot fully control the QoS requirements associated with all session set-ups, since some media simply are not associated with a CODEC (e.g. white board) or are associated with a CODEC that the proxy doesn’t know (e.g. new applications). When requesting a new service, the

CODEC in use depends on the level of QoS granted. Thereby, a higher QoS will imply the use of a CODEC that offers a higher quality. Whereas low QoS requirements are followed by a greater compression that attaches information lost.

7.3 THE SOLUTION: EQ-SIP

To express the QoS desired and to allow negotiation of QoS, it is required a two ways SIP extension:

(1) QSIP: adding QoS headers in SIP messages to allow resource reservation also from intermediate entities (i.e. SIP proxies).

(2) Enhancing SDP to support QoS negotiation between QoS aware entities and SIP entities.

⇒ The result is **EQ-SIP**

Both QoS preconditions and the proposed EQ-SIP solutions present some limitations that should be considered in order to choose the best approach, either based on existing solutions or on a completely new one. With either approach the QoS related information carried inside SIP messages mainly concerns the status of the resource reservation in the access networks.

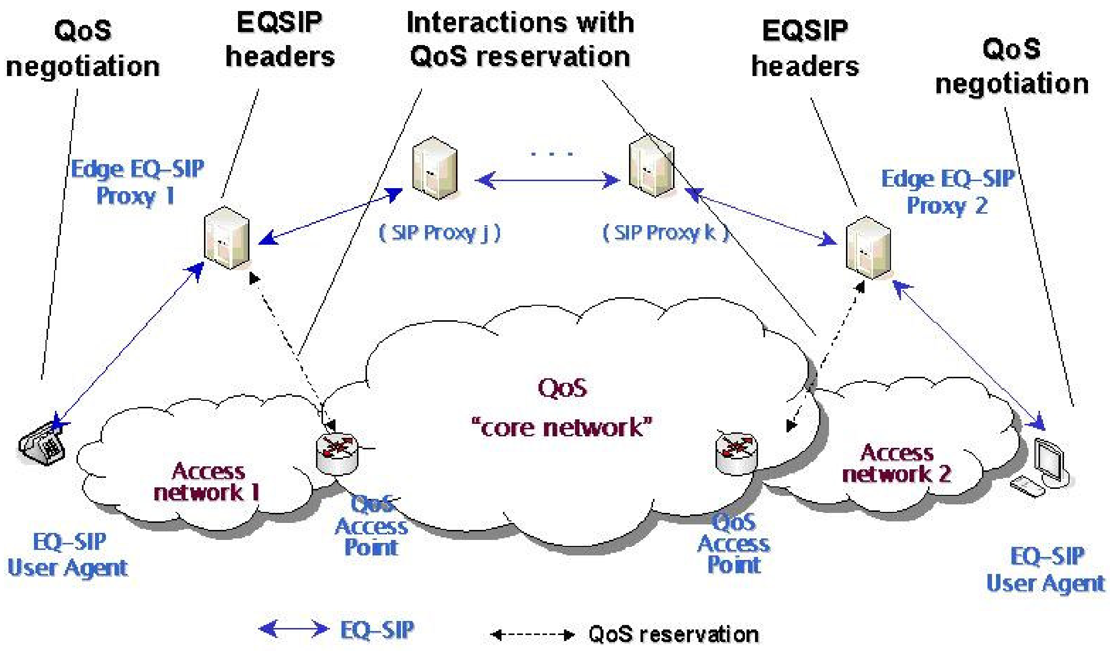


Figure 16: EQ-SIP framework

In this extended QoS aware SIP the end-user applications are EQ-SIP User Agents. They use EQ-SIP to both negotiate QoS with each other and to signal their needs to the edge EQ-SIP proxies. The Edge EQ-SIP proxy needs to exchange some information with the other EQ-SIP proxies and these proxies will request the necessary resources from the edge Resource Managers (RMs). This is shown in **Figure 17**.

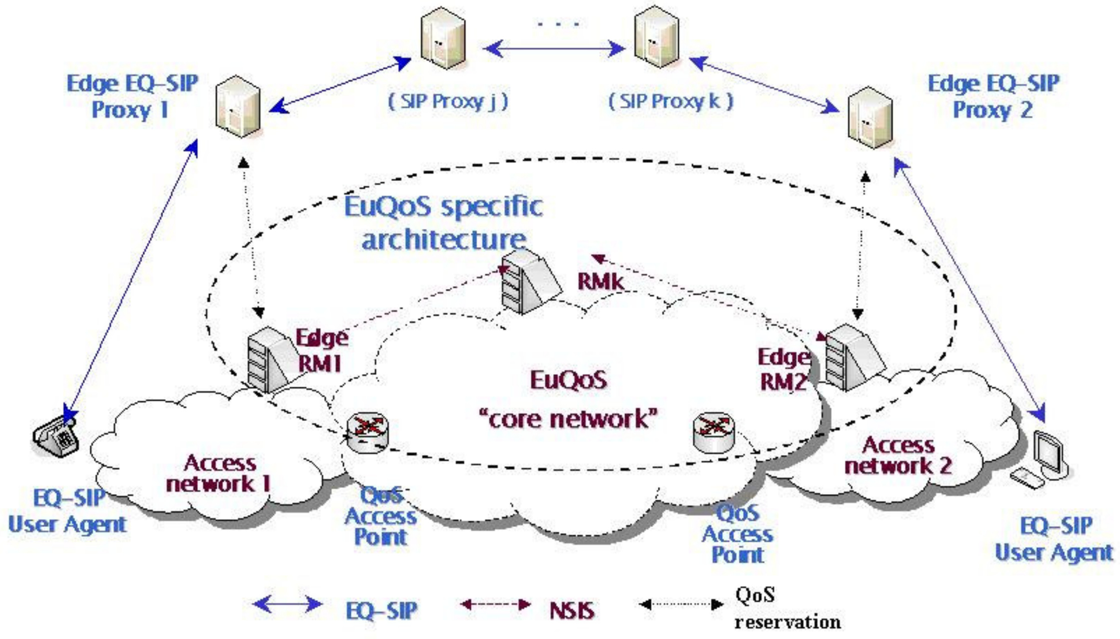
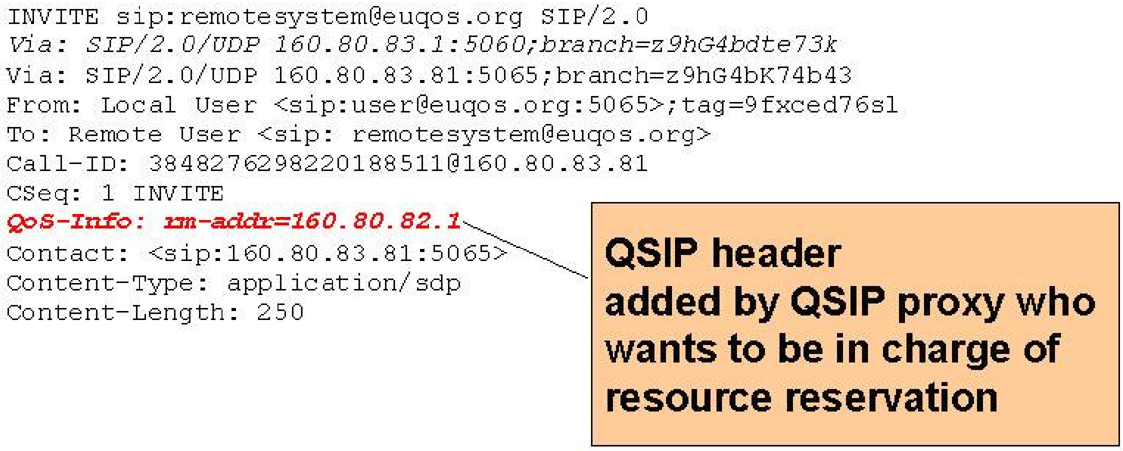


Figure 17: EQ-SIP framework & EuQoS

7.3.1 Issue 1 solution: Q-SIP

A first version of an extended QoS aware SIP was proposed in 2001 [8] [9], and updated in 2002 [10] [11]. The resulting QSIP proposal (*draft-veltri-sip-qsip-01*) [12] enhanced SIP protocol to convey QoS related information. The solution preserves backward compatibility with non-QoS aware SIP implementation and includes the handling of QoS in the SIP signaling in a flexible way. The QSIP protocol extensions needed for the QoS setup can be transparent to SIP user agents or legacy proxies. Resource reservation can be handled by user agents or by proxy servers according to the scenario. Generic mechanisms to exchange QoS information can be particularized for a specific QoS mechanism by defining specific information elements.

In the case of non-QoS aware user agents the QSIP header can be inserted by a QSIP proxy as shown in next figure.



```
INVITE sip:remotesystem@euqos.org SIP/2.0
Via: SIP/2.0/UDP 160.80.83.1:5060;branch=z9hG4bdte73k
Via: SIP/2.0/UDP 160.80.83.81:5065;branch=z9hG4bK74b43
From: Local User <sip:user@euqos.org:5065>;tag=9fxced76sl
To: Remote User <sip: remotesystem@euqos.org>
Call-ID: 3848276298220188511@160.80.83.81
CSeq: 1 INVITE
QoS-Info: rm-addr=160.80.82.1
Contact: <sip:160.80.83.81:5065>
Content-Type: application/sdp
Content-Length: 250
```

QSIP header
added by QSIP proxy who wants to be in charge of resource reservation

Figure 18: QSIP header

The **QoS-Info** new QSIP header indicates the IP address of the resource manager in charge of the resource reservation.

7.3.2 Issue 2 solution: QoS negotiation with SIP

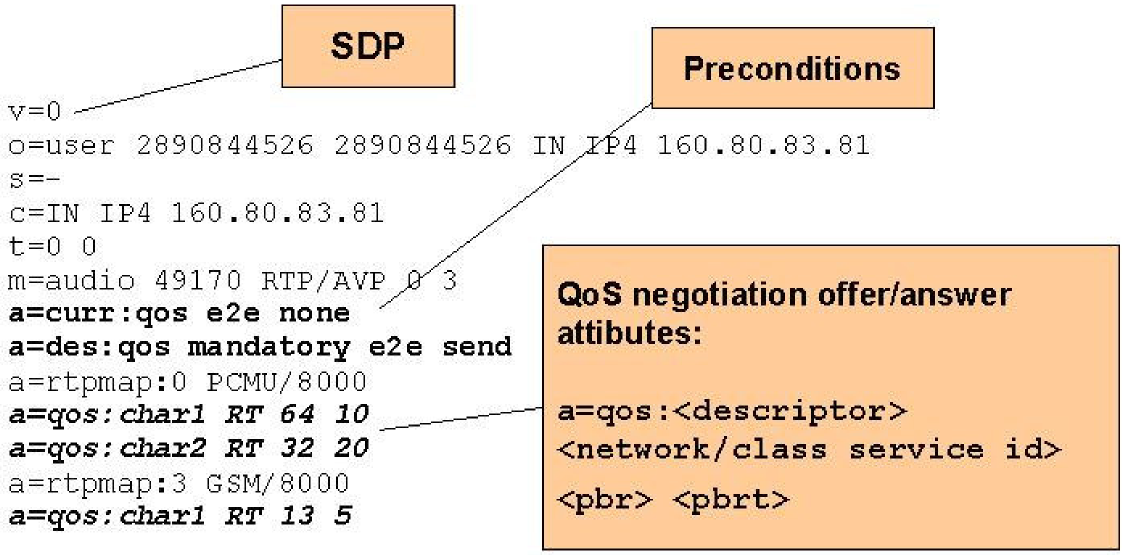
It is not possible to negotiate QoS using SIP protocol without an enhancement. It is necessary to involve SIP in the QoS negotiation defining a framework in which end-users applications negotiate QoS requirements and characteristics of the media components in a session. The EQ-SIP proxies are able to derive QoS requirements and characteristics of the applications.

The application QoS information is expressed within the SDP body of SIP messages. Since the purpose of the SDP protocol is to carry session and media stream descriptions, extensions to this protocol seem the natural way to carry this QoS information.

The QoS negotiation is modelled after the existing Offer/Answer negotiation of CODECs:

- The offer determines a set of acceptable QoS levels per media stream and, eventually, per CODEC.
- The list of acceptable QoS levels is carried in the SDP offer in decreasing order of preference, to allow prioritization during negotiation.
- The answerer is only allowed to reject or restrict the offer removing the QoS levels which are not acceptable to it.

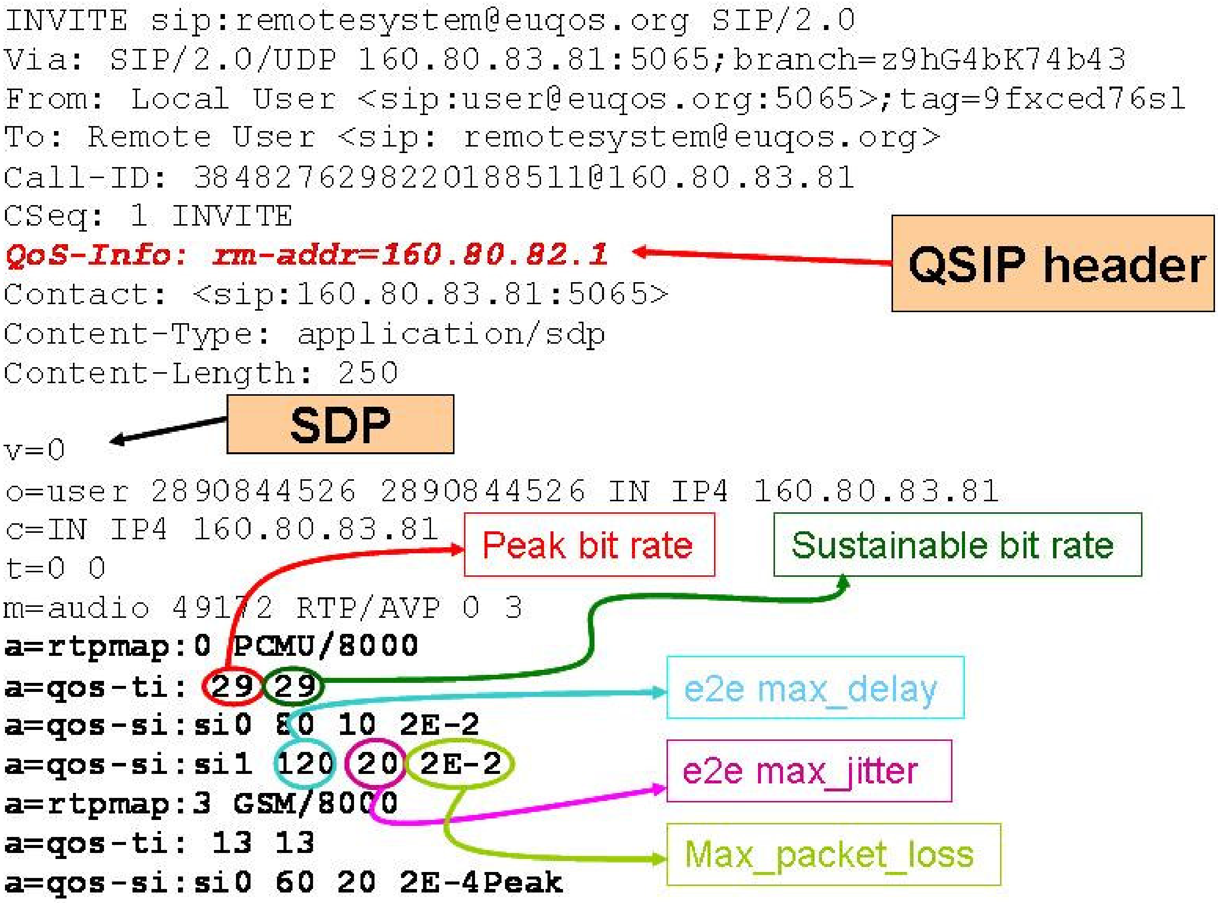
The need for actual negotiation rather than simply offer and accept or reject has already been identified in the context of SDP next generation (SDPng) [13] work within Multiparty Multimedia Session Control (mmusic) IETF (Internet Engineering Task Force, http://www.ietf.org/) Working Group. It has been proposed to support QoS negotiation in SDP next generation, with a set of SDPng extension fields. The need to implement a simplified solution for current SDP was also identified in [13].


SDP
Preconditions
QoS negotiation offer/answer attibutes:
a=qos:<descriptor> <network/class service id> <pbr> <pbrt>

```
v=0
o=user 2890844526 2890844526 IN IP4 160.80.83.81
s=-
c=IN IP4 160.80.83.81
t=0 0
m=audio 49170 RTP/AVP 0 3
a=curr:qos e2e none
a=des:qos mandatory e2e send
a=rtpmap:0 PCMU/8000
a=qos:char1 RT 64 10
a=qos:char2 RT 32 20
a=rtpmap:3 GSM/8000
a=qos:char1 RT 13 5
```

Figure 19: QoS negotiation within SDP

7.3.3 How it might look like all together?



```
INVITE sip:remotesystem@euqos.org SIP/2.0
Via: SIP/2.0/UDP 160.80.83.81:5065;branch=z9hG4bK74b43
From: Local User <sip:user@euqos.org:5065>;tag=9fxced76sl
To: Remote User <sip: remotesystem@euqos.org>
Call-ID: 3848276298220188511@160.80.83.81
CSeq: 1 INVITE
QoS-Info: rm-addr=160.80.82.1
Contact: <sip:160.80.83.81:5065>
Content-Type: application/sdp
Content-Length: 250

v=0
o=user 2890844526 2890844526 IN IP4 160.80.83.81
c=IN IP4 160.80.83.81
t=0 0
m=audio 49172 RTP/AVP 0 3
a=rtpmap:0 PCMU/8000
a=qos-ti: 29 29
a=qos-si:si0 80 10 2E-2
a=qos-si:si1 120 20 2E-2
a=rtpmap:3 GSM/8000
a=qos-ti: 13 13
a=qos-si:si0 60 20 2E-4Peak
```

Figure 20: EQ-SIP protocol

Here, the QoS request is transported in the SDP [7] message body of the SIP protocol. A SIP server in proxy mode handles this QoS SDP and sends the request to the Control Plane (i.e. the Resource Manager) to start the resource reservation.

7.4 AAA in EuQoS

Before a user should be provided with any EuQoS services, the user has to sign up in EuQoS. This could be made through an online process or even on the phone, but either way the ISP administrator introduces the user's data in the system. This process is referred as **provisioning**.

A user has to be authorized when requesting any service. The EuQoS system will authenticate each user every time that he/she wants to access a service. This is performed at session setup phase, where the local SSN contacts with an AAA server. Since the provisioning of the service has been done in advance, the AAA server authorizes the user to start the session. After this, the resource reservation is performed.

Once the session has been granted, then accounting should start. Form a business point of view, none of the network providers will permit an unpaid use of their resources. Hence, each AS keeps a track of the resources in use in their network. The A-SSN is in charge of notifying each AAA server along the resource reservation path to start the accounting procedure. AAA servers will also be notified when the session ends, to finish the accounting. Additionally, an accounting event signal is sent to the AAA server if any parameter is changed during the communication.

When all accounting information is gathered, the AAA sends a CDR (Call Detail Record or Charging Data Record) to the charging module (CHAR). This last entity will ask for the money, either to the end user who signed up the contract or to the user's Home domain, based on specific pre-established rules. This aspect is still to be determined.

One main issue that appears in the EuQoS context is WHEN to start the accounting. In traditional systems (non-QoS) the billing begins when the called party accepts the connection (i.e. picks up the phone). Before the session is established, the operators face an **unpaid** signalling cost. However, in EuQoS the signalling is not the only unremunerated expense. Several resources have been reserved in advance; therefore, there is an implicit cost to this operation. It is not only the signalling cost, but the reserved resourced cannot be used by anyone else. The AS resources are "in use" by the call even though there is not any media using these resources.

This report will also study a roaming environment. There are some new facts to consider when the user is not within their home network: Where is the user subscribed? Should he or she be provided with a service that is requested in a Visiting network? How shall it be charged? These issues will be covered later in later chapters.

Finally, there are reports where both main topics are covered: QoS and AAA, examples are: "Advanced Authentication and Authorization for Quality of Service Signaling" [16], where the Resource Reservation Protocol (RSVP) [17] is proposed as the best solution to make resource reservations in network nodes.

8 Use case description

8.1 Roaming overview

While we might initially assume that resources in use belong to the same operator; it is obvious that this point of view is not accurate in the real world. Users travel over the globe and they would like to use a service independent of their location. There is where roaming comes into play; the goal is to provide the same service that a user would enjoy in his home network.

When a subscribed user temporally visits another network, he or she may want to utilize the services that there are used to in their home network from their current location. In this case, some kind of interaction is required between the user's foreign domain and home domain. This interaction is needed, for instance, to allow a user agent to utilize services provided by a foreign services provider while the user is roaming.

The following use cases will be analyzed in this document: use case 1: Non-roaming scenario and use case 2: Roaming scenario.

Use case 1: Non-roaming scenario

a) Both communicating parties, i.e., the calling and the called party, are connected to the **same** network, where a service provider is applying EuQoS. There is only one network involved and only one EuQoS provider; therefore, roaming doesn't need to be considered. (This scenario is not considered further in EuQoS project phase 1.)

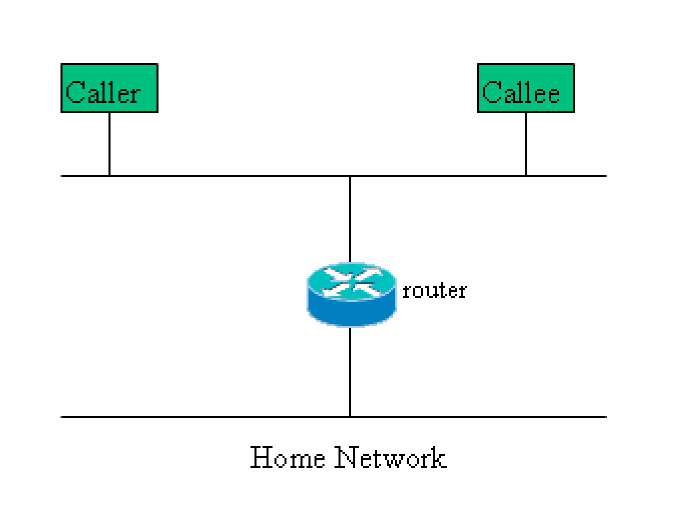


Figure 21: Non-roaming case (a)

b) A different, but still a non-roaming scenario would be when each party, i.e., the calling and the called party, are both connected to **different** networks, but each network is connected to the same network where users made their EuQoS subscription. (This scenario is to be considered in EuQoS project phase 1.)

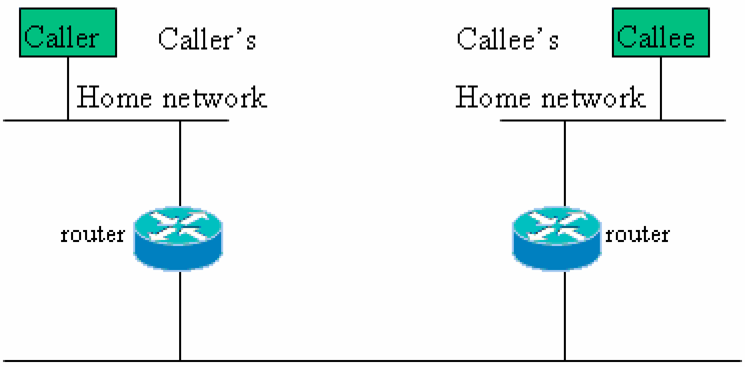


Figure 22: Non-roaming case (b)

8.1.1 Use case 2: Roaming scenario

a) The calling subscriber is **temporally visiting** a network (here after called the Visited Network), different from his or her Home Network. The calling party will try to communicate with called party using EuQoS.

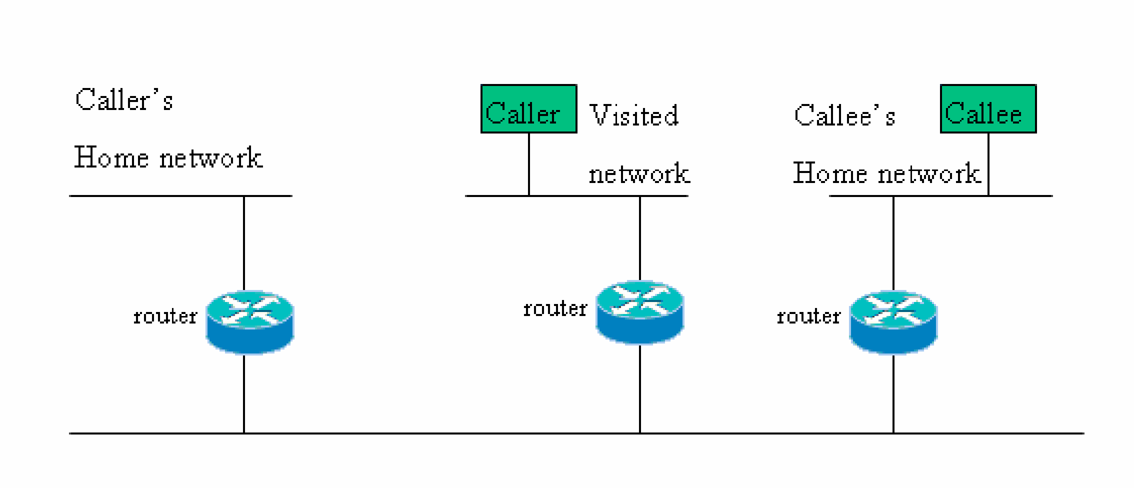


Figure 23: Roaming case (a)

b) The most general case will be when the called party is also located in a Visited Network.

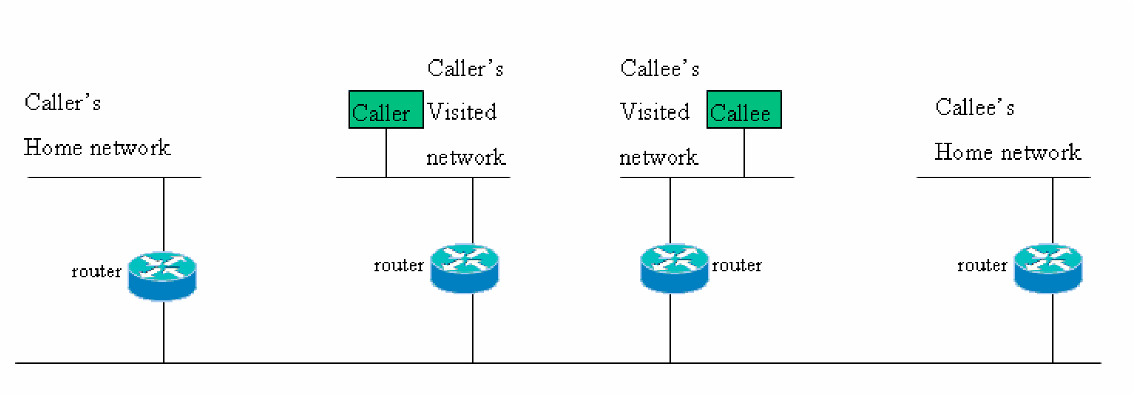


Figure 24: Roaming general case (b)

As shown in [69], one approach to support roaming is to use RADIUS [22] to carry authentication information. Standard RADIUS proxying is able to carry AAA information. Initially, RADIUS was deployed to provide dial-up Point to Point Protocol (PPP) and terminal server access. Over time, with the growth of the Internet and the introduction of new access technologies, including wireless, DSL, Mobile IP, and Ethernet, routers and network access servers (NAS) have increased in complexity and density, putting new demands on AAA protocols.

Today, DIAMETER is emerging to support this new complexity and to provide roaming services. Taking into account the increasing number of Internet service providers (ISPs) today, the ROAMOPS Working Group realized that requiring each ISP to set up roaming agreements with all other ISPs did not scale well. Therefore, the working group defined a broker, which acts as an intermediate server, whose sole purpose is to set up these roaming agreements. A collection of ISPs and a broker is called a roaming consortium. There are several such brokers in existence today and many also provide settlement services for member ISPs. This same approach could be used in EuQoS

In “Inter-domain Authentication and Authorization Mechanisms for Roaming SIP Users” [25] the authors describe two possible approaches for exchanging authentication, authorization, and accounting information between foreign and home providers: a SIP dependent and an independent inter-domain AAA communication.

In the SIP dependent scenario, SIP is used as the communication protocol between the interacting providers and for carrying any information that needs to be exchanged between the providers. With the SIP independent scenario a special AAA protocol is used between the domains for exchanging AAA related information. Both approaches are described in terms of message sequences in [25]; here I will only refer to the final conclusions extracted from that document:

8.1.2 Evaluation of a SIP dependent AAA communication

- **Complexity:** Each service provider can use their own proprietary AAA servers. All inter-domain communication is then realized based on the standardized SIP messages. That is, no separate inter-domain AAA communication is required. This helps to simplify the provider’s infrastructure. On the other hand, this increases the complexity of the SIP signalling itself. Due to this the SIP messages need to be specially protected to prevent such misuse, which increases the complexity of using SIP.

- **Performance:** Application signalling needs always to pass through the home provider even if it would have been possible to go directly to the called party without reaching the home provider’s network first. This gives the home provider a better view of the user’s actions. Thus, Authentication and Authorization of any service can be performed at the home network making possible to control the user’s access. A longer round trip delay is the immediate drawback associated to this constrain.

- **Security:** Some kind of security association between the home and foreign networks needs to be established in order to allow the foreign network to accept the AAA data generated by the home network and vice-versa. This might still need some integration of the providers with a PKI infrastructure or some security broker to dynamically establish such an association, especially when dealing with a large number of providers.

8.1.3 Evaluation of a SIP independent AAA communication

- **Complexity:** An SIP independent inter-domain communication reduces the security requirements on the SIP messages. However, the providers need to support yet another protocol and standardized components.

- **Performance:** Since now only the AAA data need to be exchanged between the foreign and home network whereas the application signalling, data can be exchanged directly between the caller and callee. This reduces the traffic load on the home provider's proxies and eases the signalling delay. Accounting data can also be exchanged over the AAA infrastructure without having the need for a separate protocol or the need for integrating the data with the application signalling protocols. In case the SIP home provider needs to be on the signalling path in order to provide the users with some services, the SIP independent AAA exchange would be wasteful. In such case, the inter-domain communication between the home and foreign providers would be generated twice: once for the SIP signalling and once for the AAA exchange.

In conclusion, you should be aware that both schemes show a substantial level of complexity. Therefore, when really introducing the one or the other, the implementers need to evaluate the exact communication circumstances and base their selection of which approach to use on that scenario.

8.2 General Accounting and Charging issues

Charging for EuQoS poses a significant challenge for the operators, as it involves an evolution from bearer charging to service charging. This is parallel to the evolution of the operator's role from bit pipe provider to service provider that is facilitated by the introduction of IP Multimedia Subsystem (IMS). Note that that if an operator makes this transition to a service provider they are no longer a common carrier, thus they become liable for the content of the traffic.

Users of EuQoS services will demand fair pricing, which implies a charging model that is simple, predictable, easy to understand, and directly related to the used service. This indicates a move away from the volume and time based charging models of today to a service based charging model related to the used service such as pay-per-picture, pay-per-message and pay-per-value.

End user pricing /business models

Different pricing models exist in the broadband and wireless market: monthly, daily subscription fees, prepaid, bucket bundling, etc. Therefore, EuQoS shall assure in its network that all pricing models can be supported.

8.2.1 Different services – different charging needs

In order to define what resources the EuQoS system needs to account for, when a resource should be accounted for, how and who will account for it, I will enumerate some examples of charging models. For the purpose of structure and common understanding, the following terminology for four different types of charging categories will be used.

- Content charging: This type of charging is characterized by charging for the actual value of the content. It is applicable when users license content for a single use or for multiple uses within a fixed period.

- Service charging: C(service). This type of charging is based on a charge per transaction of a particular service. For example, charging users per SMS or MMS.

- Bearer charging: This is the charging category that accounts for the amount of data (Mbits, Kbits) being consumed. This can be compared to consumed minutes for voice calls in the circuit switched domain, where the only differentiation of bearer is the number of minutes which the bearer is assigned to a user.

- Time based charging: C(t). This is most common used pricing model for voice call and also in the future such a model will be applied for certain conversational services.

- Flat rate: This is the most popular charging model applied to Internet access. Users don't want to bother about how much time they are online or how many bits they use. That is why flat rate is widespread. In the last years, more and more people are changing their time based telephone contracts to flat rate.

Note: There is a great difference of terminology in the market when it comes to different types of charging, e.g. application charging, event charging, session charging, volume charging, access charging, etc.

The true flexibility that is needed to bring full freedom to operators' business models requires the possibility to mix the above charging categories in any combination. This provides business and marketing freedom and flexibility that the operators need.

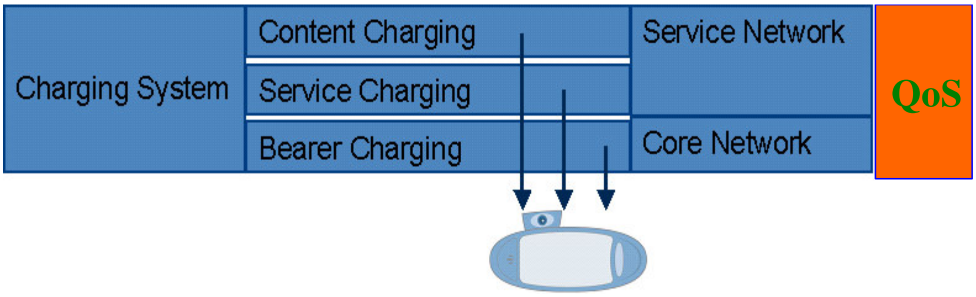


Figure 25: Charging models

Services should be charged as high up as possible in the value chain, i.e., it is like selling wood or making furniture with the wood you own and then sell the furniture at a higher price. Operators must also have the **flexibility** to select whether or not users that are buying a service on a higher level should be charged for transport in the lower levels.

The figure above can be seen as a value chain, showing that services on a higher layer in the value chain are transported through the lower layers to reach the user's terminal.

This means in theory that the operator could charge for the content the user purchases, the MMS service that is used to push the content to the user, and the GPRS bearer traffic that is generated. However, consumers will demand simplicity, lack of surprise in billing, and fair pricing, which in turn will place requirements on the operators. Hence while the operators will have to be able to price in a flexible manner, i.e. to select in which layers to charge for different services and to be able to set different prices for different services in the different layers, there will be limits to the pricing models which they can actually use in the market.

For all these scenario's correct charging must be technically implemented in a cost effective and flexible way. This requires standardization and interaction with different elements in a total network.

Moreover, there are two main charging architectures that could be used:

- *Offline charging architecture*, e.g., for post-paid contracts.
- *Online charging architecture*, e.g., for pre-paid contracts.

8.2.1.1 Off-line charging

EuQoS enables offline charging for:

- Flat rate (pay per month),
- Pay for session duration,
- Pay per event,
- Pay for event time/volume,

- A-party pays for all, or B-party pays for all,
- Everybody pays for their own part.

8.2.1.2 On-line charging (Prepaid, Credit Limit)

EuQoS enables online charging for:

- Flat rate (pay per month),
- Pay for 'floor-time' charging,
- Pay per event,
- Calling-party pays for all or called-party pays for all,
- Event & volume charging (We share image/motion + time for voice communication),
- Event charging for SIP messages.

8.2.2 Roaming and interconnect

The past has shown that new services are first rolled out in an operator's own network first. Following acceptance of these new services operators then expand the reach of these services by assuring that roaming agreements include the technical enablers for EuQoS roaming. This will assure that EuQoS services are available all over the world.

8.2.3 Charging in EuQoS

The EuQoS system does not dictate any business model. However, there are some guidelines in the solution defining the accounting approach for the EuQoS system:

- The two main actors who want to make money from the EuQoS system are:
  - Service Providers, and
  - Network Providers (note: they could be the same entity)
- If the session crosses multiple networks owned by different Network Providers, then there is a need to share the revenue with these Network Providers.
- The Service Provider charges the user, and then passes on some of the money to all the Network Providers along the route.

On-line charging during roaming is more complicated than in a non-roaming scenario. This increased complexity comes from the real-time interaction between different network operators. On-line verifications add additional delays to the service delivery, which may no longer meet the guaranteed QoS. In spite of the need for on-line and offline charging described in 4.3.1, this document assumes that an offline charging architecture will be used in EuQoS. This model has to be discussed further since it is not clear what charging model will actually be implemented.

9 Use case 1: Non-roaming scenario

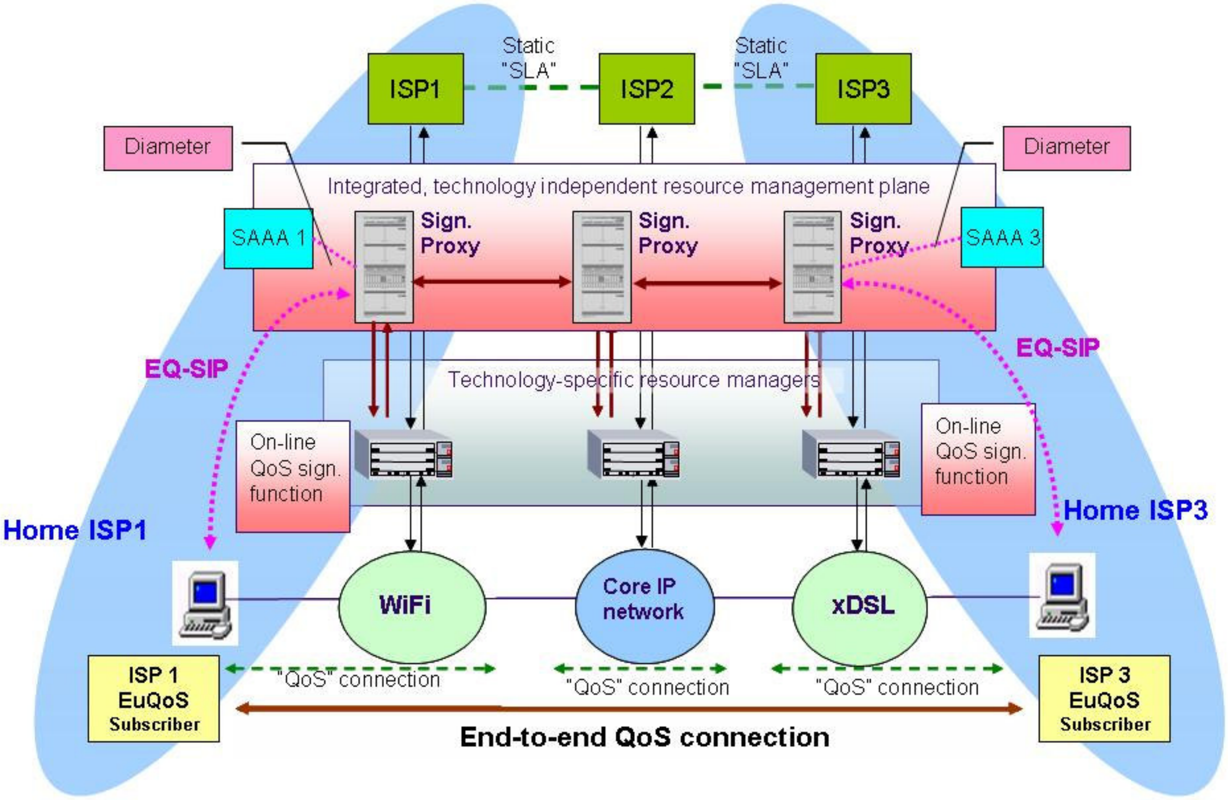


Figure 26: Non-Roaming diagram

First of all, the simplest scenario: both clients connect to each other through their Home Network. In other words, both parties, i.e., the calling and called party, are both connected to different networks, but each one is connected to the network where they have their EuQoS subscription.

9.1 Provision of EuQoS service

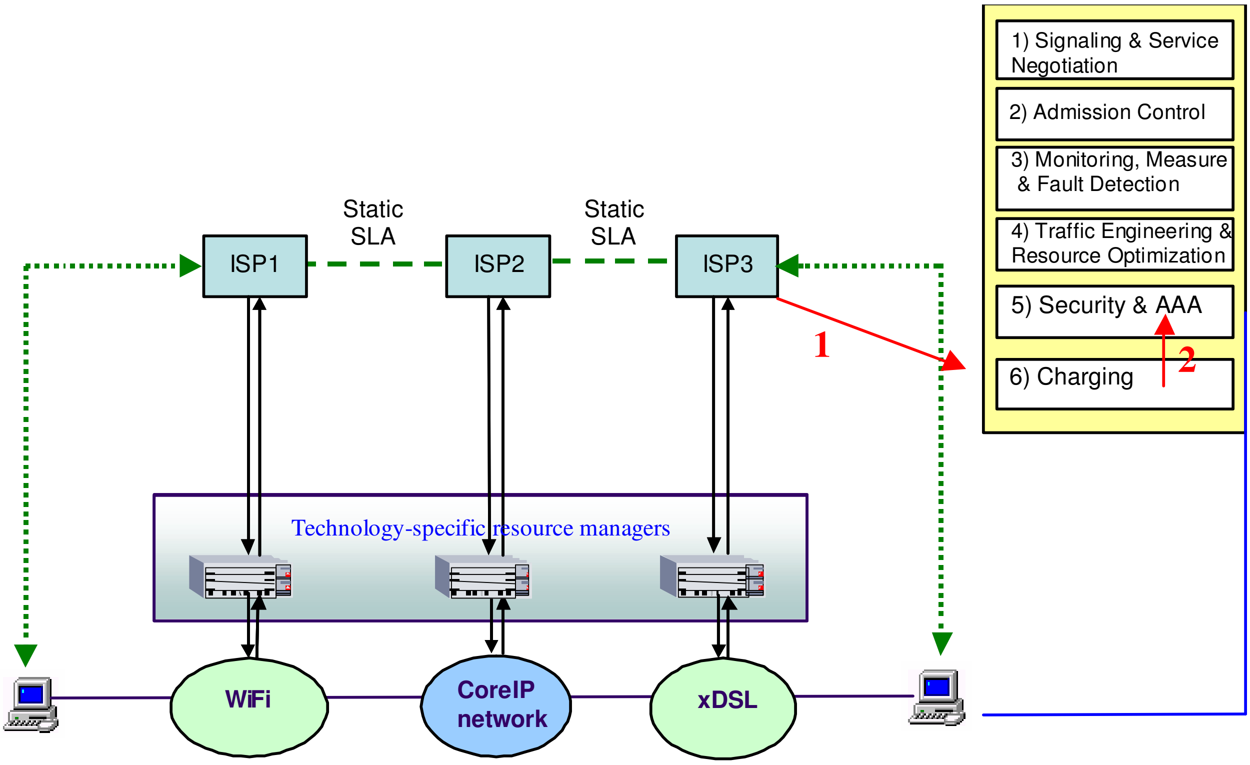

1) Signaling & Service Negotiation
2) Admission Control
3) Monitoring, Measure & Fault Detection
4) Traffic Engineering & Resource Optimization
5) Security & AAA
6) Charging
Static SLA
Static SLA
ISP1
ISP2
ISP3
1
2
Technology-specific resource managers
WiFi
CoreIP network
xDSL

Figure 27: Step by Step diagram

Step 1:

The administrator of the service provider (e.g.: ISP3) receives an order to add a new subscription with a "gold" contract (a particular level of QoS) to the EuQoS System. The administrator logins into the Charging Web Server (a web page that connects to the Charging Server), and then uses one of the options that will be available to add a new entry to the Subscription Account Data Base (at the Charging DB Server) with the requirements specified in the contract. Both Charging Web Server and Charging DB Server are located at the CHAR entity.

Step 2:

The Charging Server establishes a communication with the User Repository stored in the AAA server in order to update this DB with the information about the new subscription (EuQoS user access, EuQoS user password, and userQoSProfile).

9.2 User Registration

A user has to be authorized to establish a new communication session. Since this is a non-roaming scenario, the local SAAA server itself (the Home's SAAA) is able to authorize the user to access the user's home network. The process starts when the Application Signalling and Service Negotiation (A-SSN) node receives a new request (REGISTER, SESSION_SETUP, SESSION_RELEASE, DEREGISTER), which contains the user's credentials. Then, the A-SSN needs to ask the SAAA for authorization of the user to meet this request. Therefore, the A-SSN initiates an authentication procedure based on a challenge mechanism. This procedure involves the A-SSN, the end-system, and the SAAA, as it is shown in the following diagram.

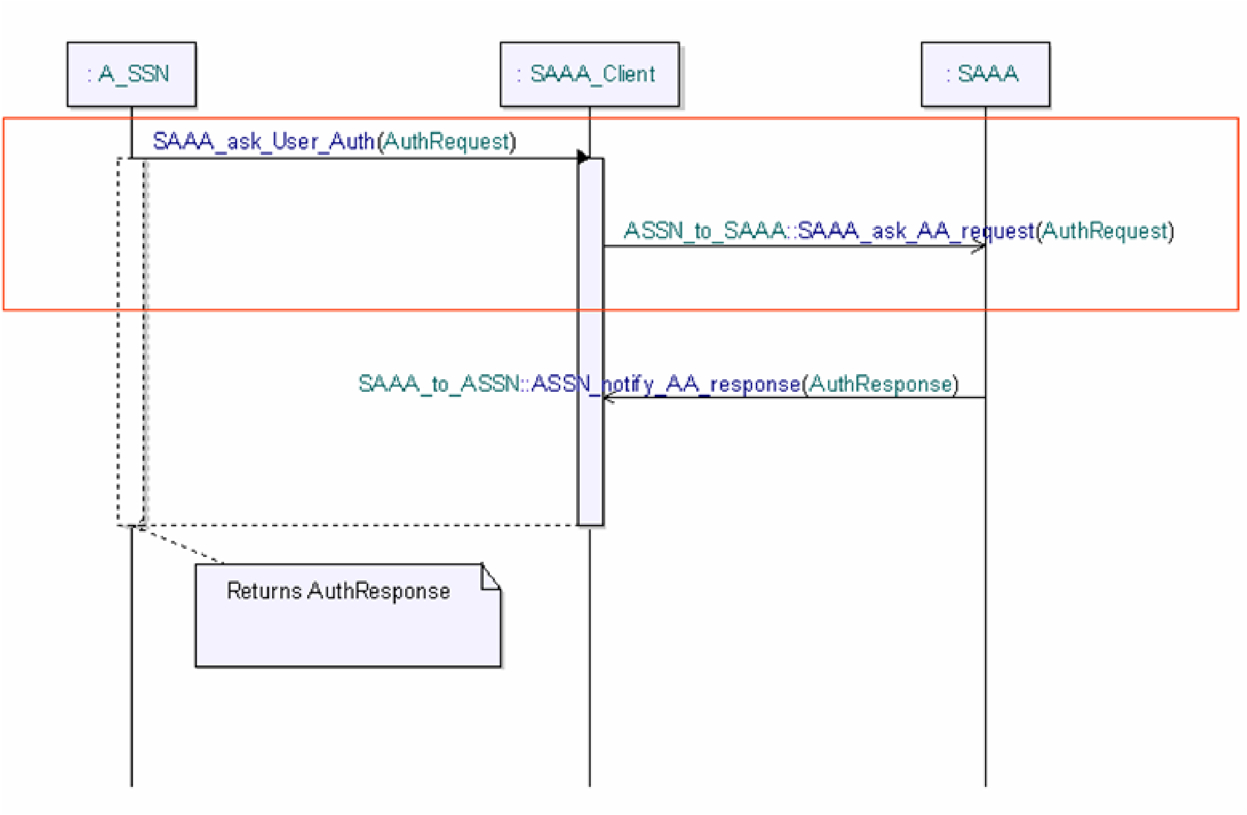

: A_SSN
: SAAA_Client
: SAAA
SAAA_ask_User_Auth(AuthRequest)
ASSN_to_SAAA::SAAA_ask_AA_request(AuthRequest)
SAAA_to_ASSN::ASSN_notify_AA_response(AuthResponse)
Returns AuthResponse

Figure 28: A-SSN requests authentication/authorization from the SAAA

Note: at the moment only the register/deregister and session setup requests need authentication, but in the future also the session release request should also be authenticated.

The data structure of the AuthResponse (Figure 32) and the AuthRequest (Figure 33) will be described in the next section (see 9.2.2).

The SAAA answers the A-SSN authorization/authentication request:

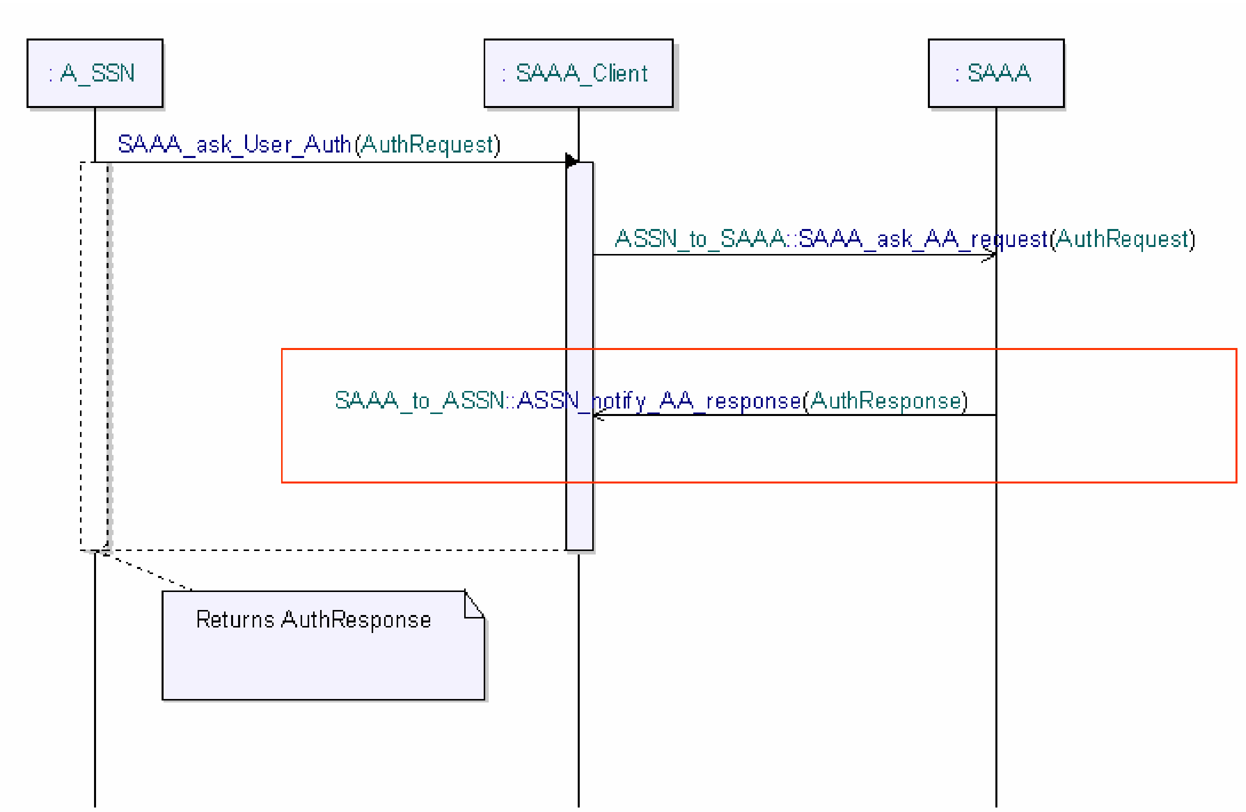

: A_SSN
: SAAA_Client
: SAAA
SAAA_ask_User_Auth(AuthRequest)
ASSN_to_SAAA::SAAA_ask_AA_request(AuthRequest)
SAAA_to_ASSN::ASSN_notify_AA_response(AuthResponse)
Returns AuthResponse

Figure 29: SAAA notifies A-SSN the AA response

The SIP registration should be made each time the User Equipment (UE) changes its IP [3] transport address. This would happen when a UE connects via a new visited domain. However, SIP registrations will happen regularly from the UE even though no change of IP address has occurred, in order to maintain its registration, which otherwise would periodically expire. This refresh mechanism is used in SIP to maintain a UA's registration.

9.2.1 User de-registration

The user de-registration is a process initiated by the user at any time. The application signaling (A-SIG) informs the A-SSN of this registration by means of the signal "SSN_notify_user_registration". After this, the A-SSN communicates with the Home SAAA, just as in the registration case.

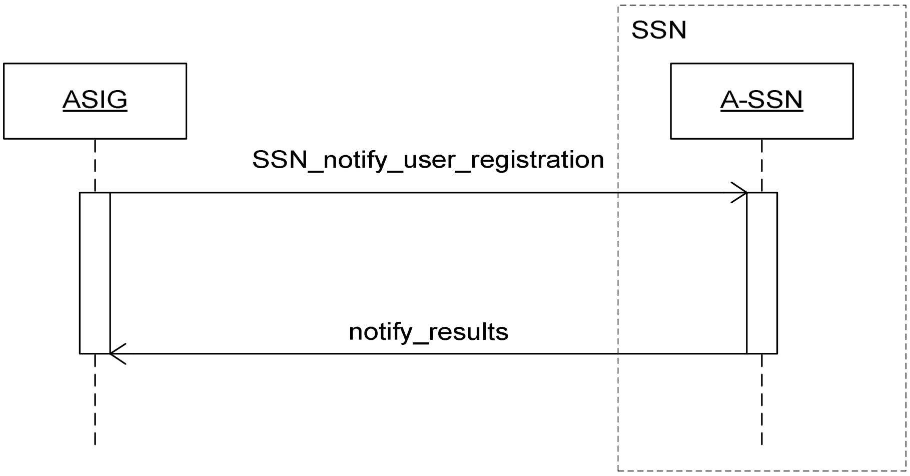


Figure 30: User de-registration sequence diagram

9.2.2 Data structures provided by A-SSN

Next figure shows the interfaces between A-SSN and SAAA. The method taxonomy used is: <name of the function> (<type of object sent as parameter>):<type of return object>

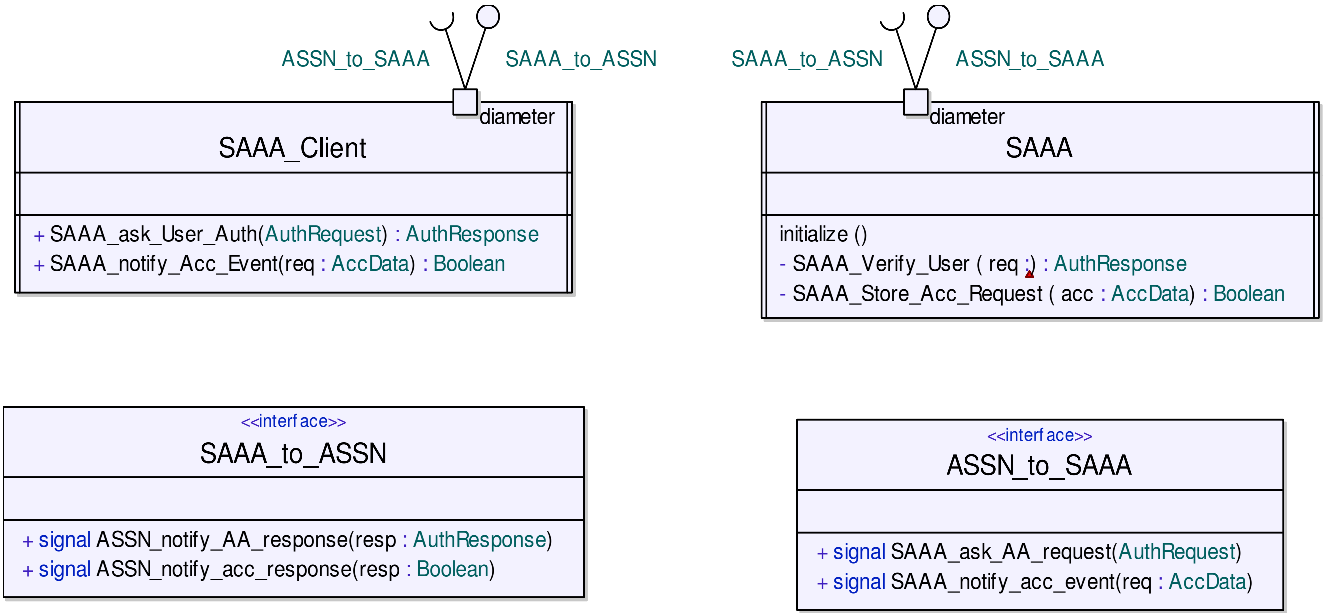

ASSN_to_SAAA
SAAA_to_ASSN
diameter
SAAA_Client
+ SAAA_ask_User_Auth(AuthRequest) : AuthResponse
+ SAAA_notify_Acc_Event(req : AccData) : Boolean
SAAA_to_ASSN
ASSN_to_SAAA
diameter
SAAA
initialize ()
- SAAA_Verify_User (req :) : AuthResponse
- SAAA_Store_Acc_Request (acc : AccData) : Boolean
<<interface>>
SAAA_to_ASSN
+ signal ASSN_notify_AA_response(resp : AuthResponse)
+ signal ASSN_notify_acc_response(resp : Boolean)
<<interface>>
ASSN_to_SAAA
+ signal SAAA_ask_AA_request(AuthRequest)
+ signal SAAA_notify_acc_event(req : AccData)

Figure 31: Interfaces between A-SSN and SAAA

The following objects are sent by the A-SSN to the SAAA in all events of session register, session setup, session release, and deregister.

The A-SSN makes a request to the SAAA for a user authentication/authorization and the SAAA answers with an AuthResponse object, which is structured as shown in Figure 32.

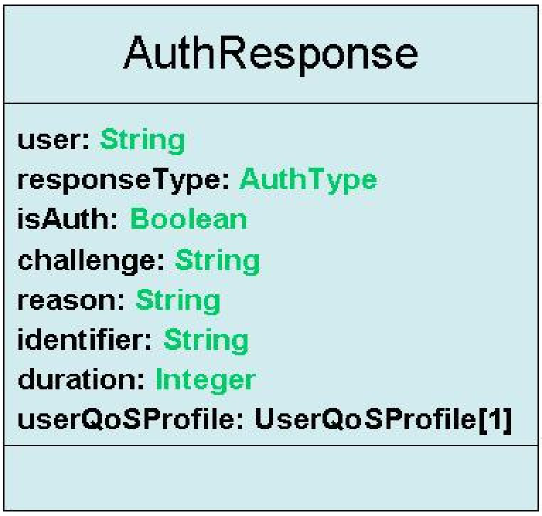


Figure 32: AuthResponse data structure

The main attributes of the AuthResponse are:

- *user*: the EuQoS username the user wants to register/start a new session.
- *responseType*: REGISTER, SESSION_SETUP, SESSION_RELEASE, DEREGISTER
- *isAuth*: a Boolean type indicating the result of the authentication/authorization process.
- *challenge*: generated by the A-SSN for this request/response and sent to the A-SIG client.
- *reason*: this parameter is filled in by the A-SSN and it provides the reason for the authentication request. For example, if the authentication has been completed successfully then this value will be a SIP “200 OK”, otherwise it will be one of the following values: “401 Unauthorized”,”407 Proxy-Authentication Required”, or “403 Forbidden”. This parameter is empty in the request from the SAAA to the A-SSN.
- *identifier*: in case of a REGISTER responseType - the value it is the euqosRegisterId which is unique within the EuQoS system. In the case of a SESSION responseType the value is the euqosSessionId which uniquely identifies the session within the EuQoS system.
- *duration*: this parameter is filled by the A-SSN during the registration phase (see [6] section 10.2.1.1) and it represents the duration of the

authentication/authorization process. After the expiration of this interval, the A-SIG must initiate a new registration procedure in order to refresh its registration status. This parameter is empty when is provided to A-SSN by SAAA.

- *userQoSProfile*: This object is a container of QoS descriptors directly derived from the user subscription contract information kept by the SAAA. The value of this object depends on the context in which the authResponse is sent:

    - In the case of a responseType="REGISTER" this object contains the description of all applications with the related QoS allowed for this user (contract) in terms of ConnectionCharacteristics as described in next subsection. The provisioning of this information is reserved for future purposes and can be inserted by the A-SSN into a session record in order to advertise constraint/allowed QoS parameters to reserve in case of session establishment. For instance, should the user subscribe to the "gold" VoIP service with an option stating that he/she may accept a lower quality in case of network congestion, this object will contain the full set of available Class of Services (for example Gold, Silver, Bronze, …). On the other hand, should the user subscribe for the "gold" VoIP service without such an option, then only the gold Class of Service would be presented to the customer. In this latter case the network should not accept the session or the session down if the resources to be allocated do not match the constraints.

    - In the case of a responseType="SESSION_SETUP", this object contains the description of QoS allowed by user contract only for the application requested by the user.

    - In the case of either responseType="SESSION_RELEASE" or "DEREGISTER" this object is empty.

During the phases of REGISTRATION, SESSION_RELEASE, and DEREGISTER authentication phases the ConnectionCharacteristics parameter is not specified at all. Instead during the SESSION_SETUP authorization phase it is mandatory to specify the ConnectionCharacteristics parameter (it is provided by A-SIG). The Authorization Request data structure looks as follows:

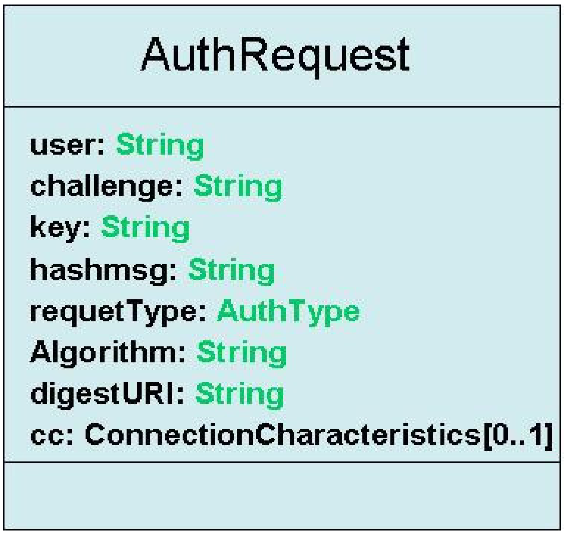


Figure 33: Authorization Request data structure

The next attributes are included in the AuthRequest object:

- *user*: the EuQoS username the user wants to register/start a new session.
- *RequestType*: the next values are allowed, as Figure 34 shows: REGISTER, SESSION_SETUP, SESSION_RELEASE, or DEREGISTER

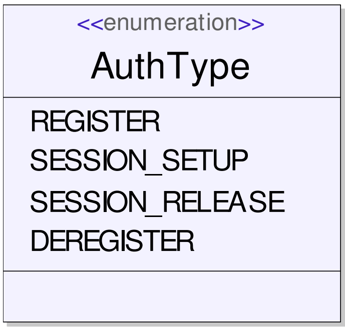


Figure 34: Authorization Type

- *challenge*: generated by the A-SSN and sent to the A-SIG client.

It is important to notice that in this application three players (A-SIG, A-SSN, and SAAA) act, the A-SSN is in charge of generating the challenge and sending it to the A-SIG; since the SAAA is in charge of the authentication/authorization process it has to receive the challenge as well.

- *key*: result of the MD5[3] function: f(H(MESSAGE), PASSWORD, CHALLENGE)

- *hashmsg*: the hash function of the entire message H(MESSAGE)

- *algorithm*: produces the key and the challenge. If the algorithm is not understood by the A-SSN and the SAAA, then the challenge provided by the ASIG should be ignored

- *digestURI*: the URI included in the Authorization Request Header (or Proxy Authorization Request Header) sent by the ASIG in a proper request message

- *cc*: This field is an instance of *ConnectionCharacteristics* and is present only for a requestType="SESSION_SETUP" and is the same instance of *ConnectionCharacteristics* that A-SSN receives from user when it requests to establish an application session. In all other cases (REGISTER, SESSION_RELEASE, DEREGISTER) this field is empty.

9.2.3 Data structure maintained in the A-SSN

After the registration phase, when an end-system makes a new session setup request, the A-SSN internally maintains a data structure in order to describe and maintain up to date connection characteristics based on information sent through EQ-SIP messages by end-systems or a peer A-SSN. This data structure is used when the A-SSN has to cooperate with the SAAA for authentication purposes and with the Resource Manager Service and Signalling Negotiation node (RM-SSN) for the resource reservation. This data structure is very similar to the one provided by the ASIG and it is shown in next subsections.

9.2.3.1 ConnectionCharacteristics

This class contains all the connection information requested by the application. The following structure describes information related to parameters and QoS characteristics of the session that the caller is willing to establish.

- *sourceIP*: IP [3] address of the media flow IP packets that will be sent out from the calling party to the remote end-system (callee).

- *destinationIP*: IP address of the media flow IP packets that will be sent out from the called party to the calling party(caller).

[3] The Message-Digest Algorithm (MD5) was originally developed by Professor Ron Rivest of the Massachusetts Institute of Technology. This algorithm takes as input a message of arbitrary length and produces as output a 128-bit "fingerprint" or "message digest" of the input. It was conjectured that it was computationally infeasible to produce two messages having the same message digest, or to produce any message having a given pre-specified target message digest. However this is not true and MD5 is deprecated. More information on the algorithm can be found in RFC 1321.

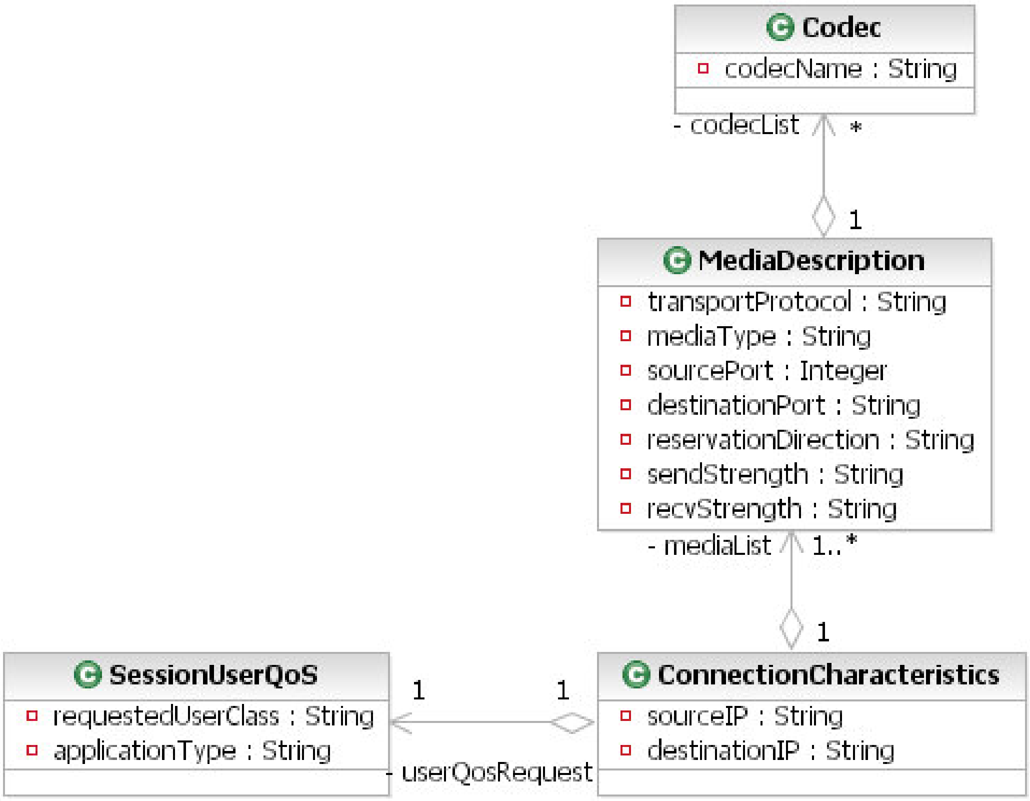


Figure 35: Data structure maintained in the A-SSN

9.2.3.2 SessionUserQoS

This class contains the user QoS request for the session establishment. When users want to establish a session, they have to choose a user class according to their account grants and provide the type of the application that is trying to establish a session. The main attributes of this object are:

- *requestedUserClass*: logical name of the user class requested for the session. This logical name can take any of the values: “gold”, “silver”, “bronze”, and “premium”.

- *applicationType*: describes the type of the application. This parameter can take a value such as: “VoIP”, “VoD”, “VideoConference”, etc.

9.2.3.3 MediaDescription

This structure contains information for a specific mediaType (“audio”, “video” or “data”) the caller wants to negotiate. Its attributes are:

- *mediaType*: the type of the media (“audio”, “video” or “data”).

- *transportProtocol*: transport protocol name. For instance: “UDP”, “TCP”, “RTP/AVP”, “SCTP”, etc.

- *sourcePort*: source port of the media flow IP packets that will be sent out from the calling party to the remote end-system (callee).

- *destinationPort*: destination port of the media flow IP packets that will be sent out from the called party to the calling party(caller).

- *reservationDirection*: the direction in which network resources are currently allocated for this particular media stream. This parameter can take one of the following values: “none”, “send”, “recv”, or “sendrcv”.

- *sendStrength*: whether the allocation of the network resources in the direction from the caller to the callee is a prerequisite for the establishment of the session. Possible values are “mandatory”, “optional” or “none”. A value equals to “mandatory” indicates to the remote peer application/end-system and to the EuQoS system (i.e. the ASSN function) that the resource allocation must be provided otherwise the session establishment must not continue. A value equals to “optional” indicates that the resources allocation should be provided but the session establishment can continue regardless of whether or not this provision is possible. A value equals to “none” indicates that no resources allocation is desired.

- *recvStrength*: whether the allocation of the network resources in the direction from the callee to the caller is a prerequisite for the establishment of the session. Possible values are “mandatory” or “optional” or “none”.

- *codecList*: list of CODEC for a given mediaType to negotiate with the remote end-system application (callee application).

9.2.3.4 CODEC

This structure contains information for a specific mediaType for instance “audio”, “video”, etc. the caller wants to negotiate.

- *codecName*: the CODEC name. For instance: “MJPG”, “GSM”, “PCMU”, “PCMA”, etc.

- *qosCharacList*: this parameter is optional (see picture below). List of QoS characteristics to negotiate with the remote end-system application (callee application) for this CODEC.

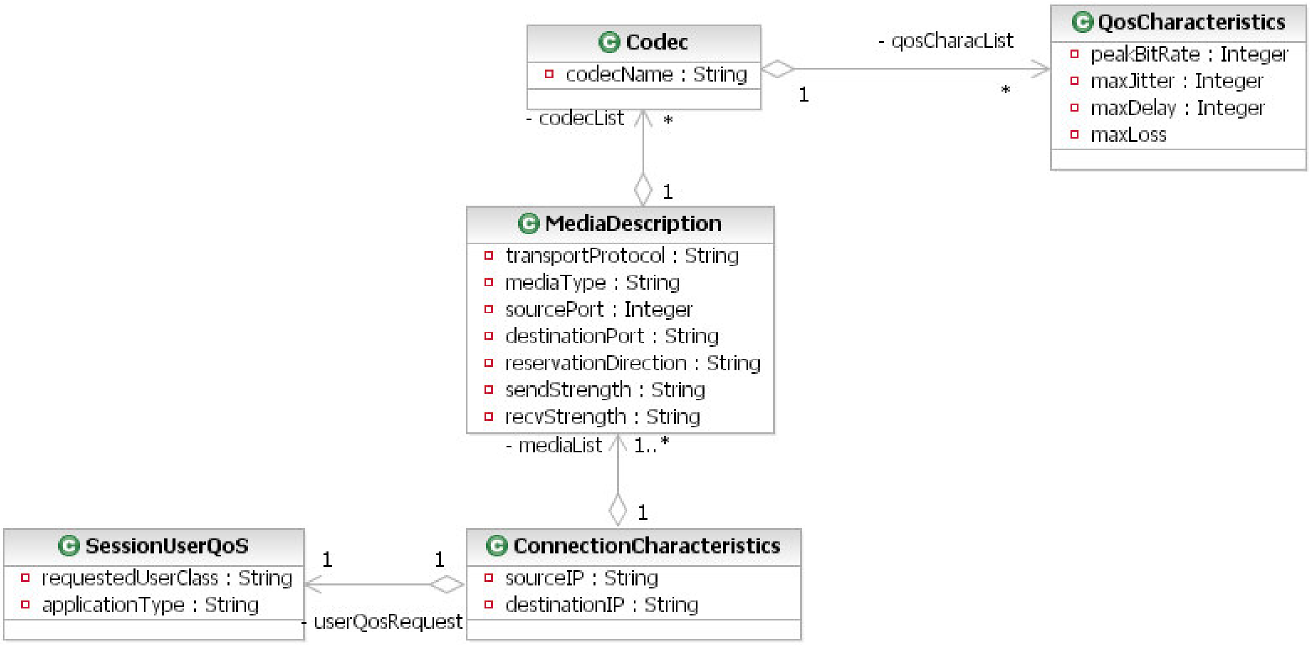


Figure 36: ConnectionCharacteristics with QoS Characteristics

If the application requires specific QoS Characteristics for the CODECs, the application will provide a ConnectionCharacteristics object with the structure shown below.

9.2.3.5 QosCharacteristics

This structure contains the QoS characteristics for a given CODEC. It is specified by the next set of parameters:

- *peakBitRate*: the requested bandwidth
- *maxJitter*: maximum jitter
- *maxDelay*: maximum delay
- *maxLoss*: maximum packet loss

9.2.3.6 QoSProfile

This structure contains information for a specific QoSProfile. Each access provider will define a specific set of QoSProfiles described by the next set of attributes:

- *userClass*: logical name of the user class. This logical name can take value such as: “gold”, “silver”, “bronze”, “premium”, etc.
- *accessProviderName*: name of the user’s access provider

- *networksQosClasses*: list of the Network QoS Classes proposed by the access provider for a given userClass.

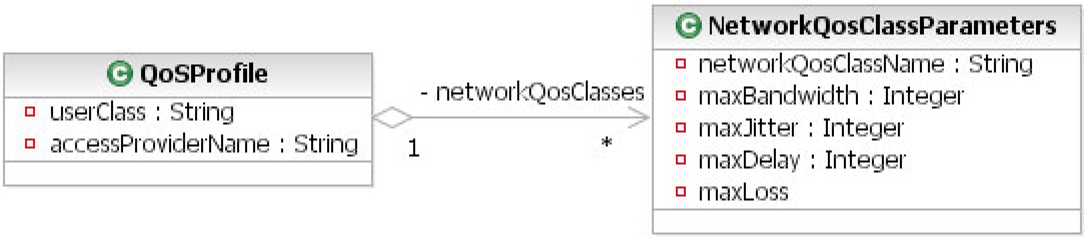


Figure 37: QoSProfile data structure

9.2.3.7 NetworkQosClassParameters

This structure contains the parameters for a specific network QoS class provided by the access network for a given *userClass.*

- *networkQosClassName*: logical name of the network QoS class.
- *maxBandwidth*: maximum bandwidth granted by the access provider for a given Network QoS Class and a given *userClass* subscription. The value depends on the access provider: “gold” for a certain provider has not necessarily the same meaning as “gold” for another provider. If a user subscribes to a “gold” profile, which means for his/her provider 100 Mbit/s: the user can request 90 Mbit/s, but the provider will reject any request above 100 Mbit/s.
- *maxDelay*: maximum delay.
- *maxJitter*: maximum jitter.
- *maxLoss*: maximum packet loss.

Authentication/Authorization: Data Flow Diagram (DFD)

A DFD is a modelling tool which represents a system as a network of process, showing the flow of information from one process to another. It is commonly used in systems analysis when designing a software system. However, it can be used to analyze any process where there is a flow of information. DFDs are stateless as they only describe where data goes, not when or in what sequence.

The following diagram is a DFD picture of the SAAA system regarding Authentication/Authorization features.

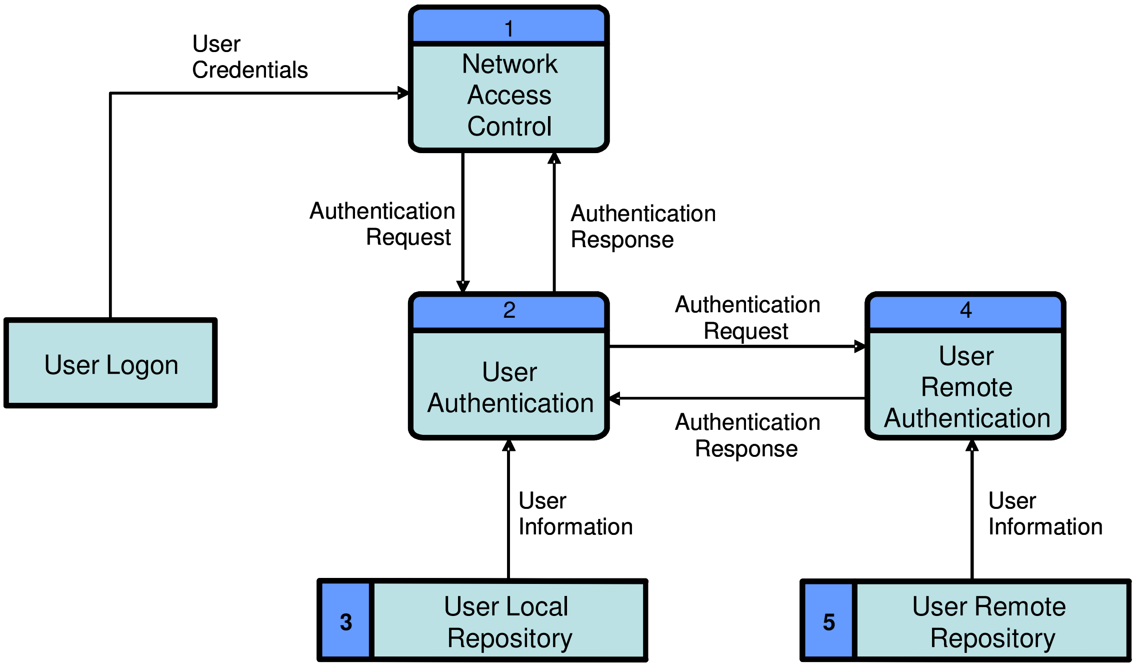


Figure 38: Authentication/Authorization DFD

9.2.4 SAAA

The following subsections describe the SAAA system and the SAAA interface. The modelling language used is the Unified Modelling Language (UML), but Specification and Description Language (SDL) and Data Flows Diagram (DFD) are used as well. SDL is used because it is a specific language adopted in the definition of Telecommunication systems and standardized by ITU-T (ITU-T Z.100). DFD is used simply to complete the overall architecture view.

SDL Diagram for the whole SAAA system

The following SDL Diagram shows the message communication inside and outside the SAAA system. It shows the interactions and protocols used between SSN, SAAA, User Repository, CHAR, and Accounting DB. Yellow boxes represent EuQoS entities, the protocol used in the communication between the entities is coloured in blue, whereas the action performed is written in black and between square brackets. The user agent uses SIP to communicate with the SIP-Proxy, which interacts with the AAA server by means of Diameter. Both the ‘User DB’ and ‘Accounting DB’ are located at the ‘DB Server’.

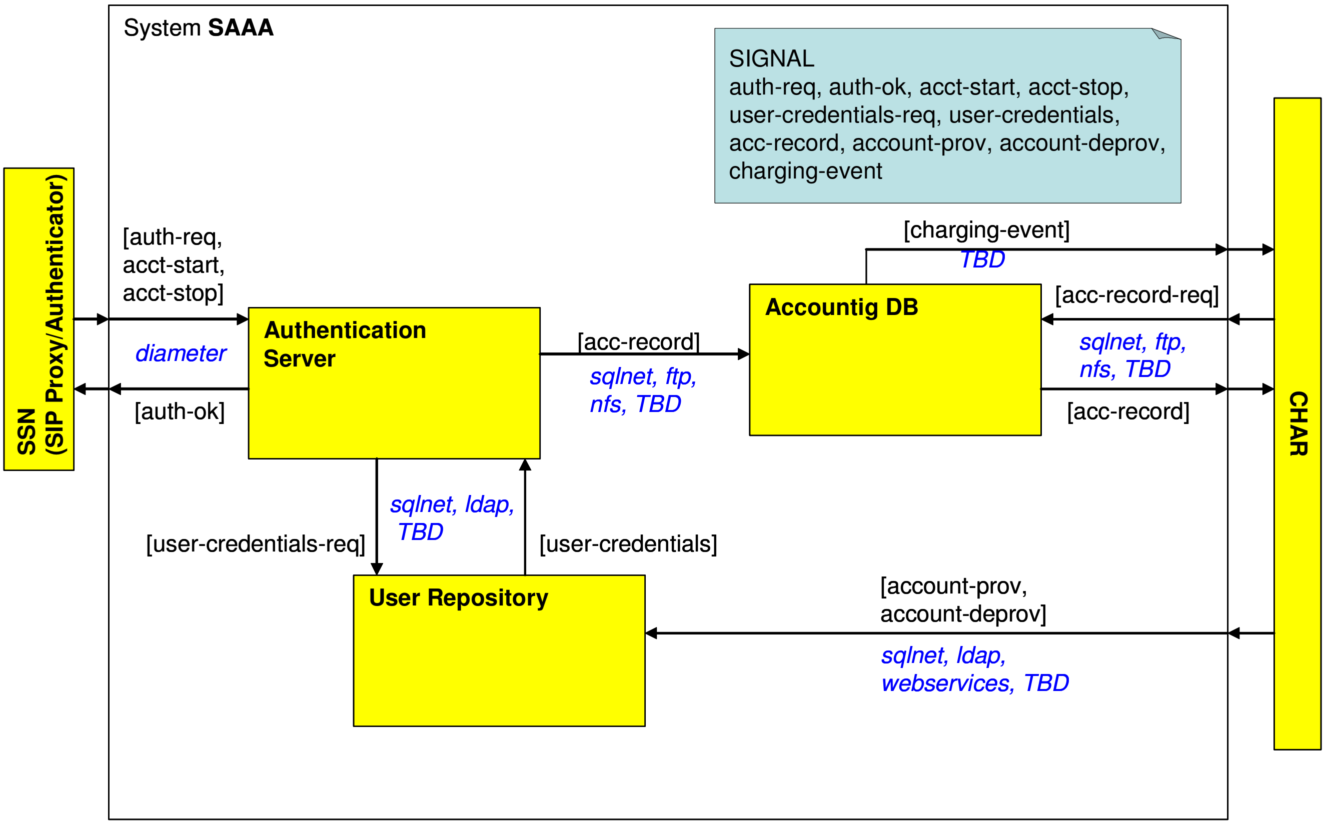


Figure 39: SAAA SDL diagram

9.2.5 SAAA interface

The taxonomy used is:

<return_results> <modulename>_<notify|ask|get|set>_<function_name>(<list of parameters>)

In which:

<return_results>	the values returned from a interface call;
<modulename>	the name of the module that exposes the interface;
<notify|ask|get|set>	the action requested to the module;
<function_name>	the name given to an exposed interface;
<list of parameters>	the list of parameters given to interface call.

SAAA Interface	**Activated By**	**Interface Description**
AuthResponse **SAAA_ask_AA_request** (authentication_authorization_parameters)	**SSN**	SSN asks for authenticate and authorize EuQoS subscribers.
AccData **SAAA_notify_acc_event** (accounting_information)	**SSN**	SSN notifies an accounting event
Notify_results **SAAA_notify_session_event** (session_event_information)	**SSN**	SSN notifies a session event to SAAA
Notify_results **SAAA_notify_agreed_session** (session_agreed_information)	**SSN**	SSN notifies an agreed session information event to SAAA
accounting_records **SAAA_get_accounting_records** (retrive_request)	**CHAR**	CHAR asks for accounting records related to subscribers
Notify_result **SAAA_notify_account_management** (account_details)	**CHAR**	CHAR notifies SAAA to create, modify and delete an account for subscriber

Table 1: SAAA interface

9.3 Session setup

9.3.1 High-level view of session establishment

Figure 40 depicts a possible message flow for session establishment between two applications. This figure describes the messages exchanged at a high abstraction level, avoiding the details concerning the types of messages, the values of the various fields, etc. The two end-users are attached to access networks AN1 and AN2, belonging to the same AS or to different ASs. In each access network an application, that plays the role of the caller/callee, and an EQ-SIP proxy are located. The application side is named ASIG and the EQ-SIP proxy is referred as A-SSN, which is the sub-module directly involved in the EQ-SIP signalling by the network-side. In addition, there is a Resource Manager (RM) in each AS. The different ANs, ASIGs, A-SSNs, and RMs, will be referred to by a number corresponding to the network to which they are attached (i.e., 1 or 2).

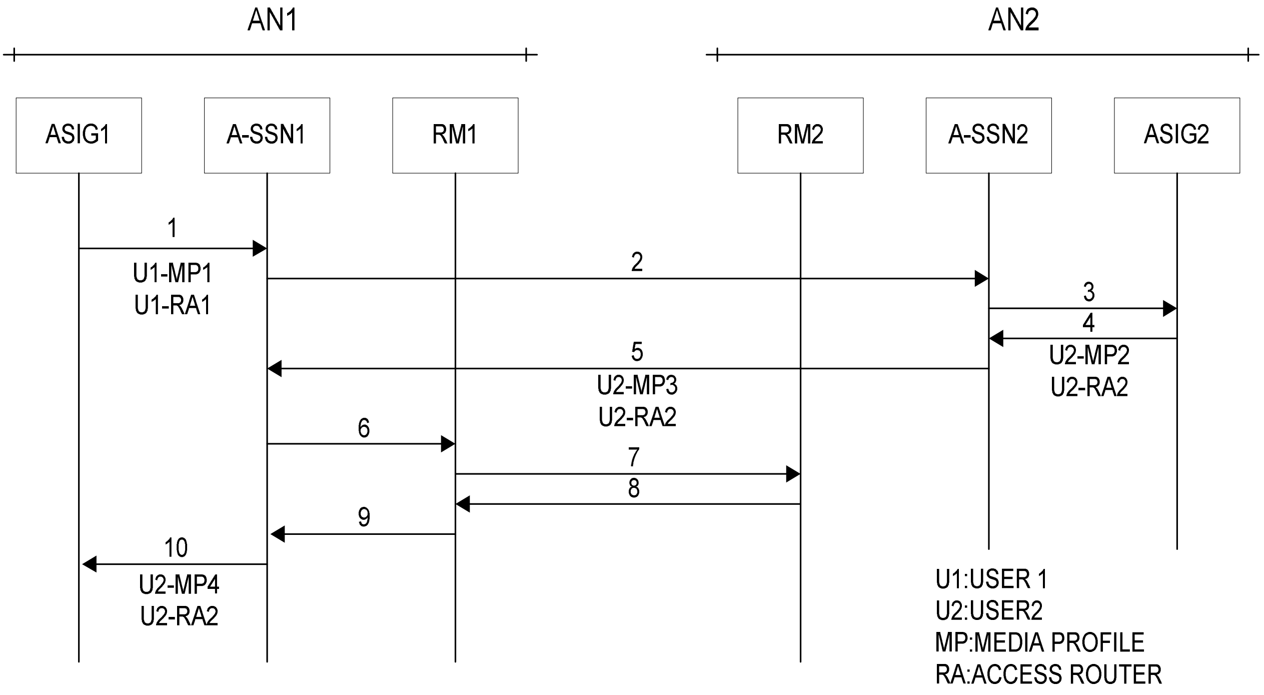


Figure 40: High-level sequence diagram of session establishment

A brief description of each message is presented below:

1) ASIG1 provides to A-SSN1 the user media profile (which includes the list of available CODECs for the specific media) and the IP address of an access router, which it has discovered, located in access network 1; Note that there may be multiple access routers for reliability and load sharing purposes.

2) A-SSN1, located in access network 1, forwards the message to A-SSN2, located in access network 2;

3) A-SSN2 forwards the message to ASIG2;

4) ASIG2 replies to A-SSN2 with the same or different user media profile (which includes the same list or a subset of the CODECs provided by the ASIG1) and with the IP address of the access router located in access network 2;

5) A-SSN2 replies to A-SSN1 with the user media profile provided by the ASIG2.

6) A-SSN1 asks RM1 for an available path, with the required QoS level from access router 1 to access router 2;

7) After a positive local check, the request is carried forward to RM2 or, in general, to the RM of the next AS;

8) RM2 replies to RM1 with a confirmation message;

9) After reception of the confirmation message from the next RM in the path (RM2, in this case) RM1 send acknowledge to A-SSN1. Note that if the

resources are not available, then RM1 tries an alternative path. The RM informs to A-SSN if it is not possible to find any other path able to fulfil the required QoS;

10) A-SSN1 forwards the confirmation message to ASIG1.

Therefore the QoS parameters negotiation is performed by ASIG1 and ASIG2, located in the corresponding access networks. A-SSN only supports the forwarding of the request/response messages based on the EQ-SIP protocol, modifying the user media profile according to the information provided by the underlying RM.

From the application and the communication system points of view, the main requirements of A-SSN signalling functions are:

- SIP-based user/application authentication/authorization;
- SIP-based user registration;
- Session establishment or update;
- Verification of the possibility to setup the connection with the required (or available) quality requirements (by interacting with Resource Manager);
- Session release.

As stated above, each A-SSN also interacts with a local Resource Manager (RM) in order to learn if a path with the required QoS is available or not. The RM receives from the A-SSN all the information required to perform the reserve of the resources, i.e., the A-SSN informs the RM about the requested QoS.

Furthermore, an interface between A-SSN and the SAAA module should be provided in order to perform user authentication/authorization.

The RM contacts with the Traffic Engineering & Resource Optimization (TERO) module to check what the best path is. After this, the RM sends a resource reservation to the Connection Admission Control (CAC) module. When CAC informs the RM about the real available resources, the RM makes the reservation by means of the Resource Allocator (RA). This process is shown in next page.

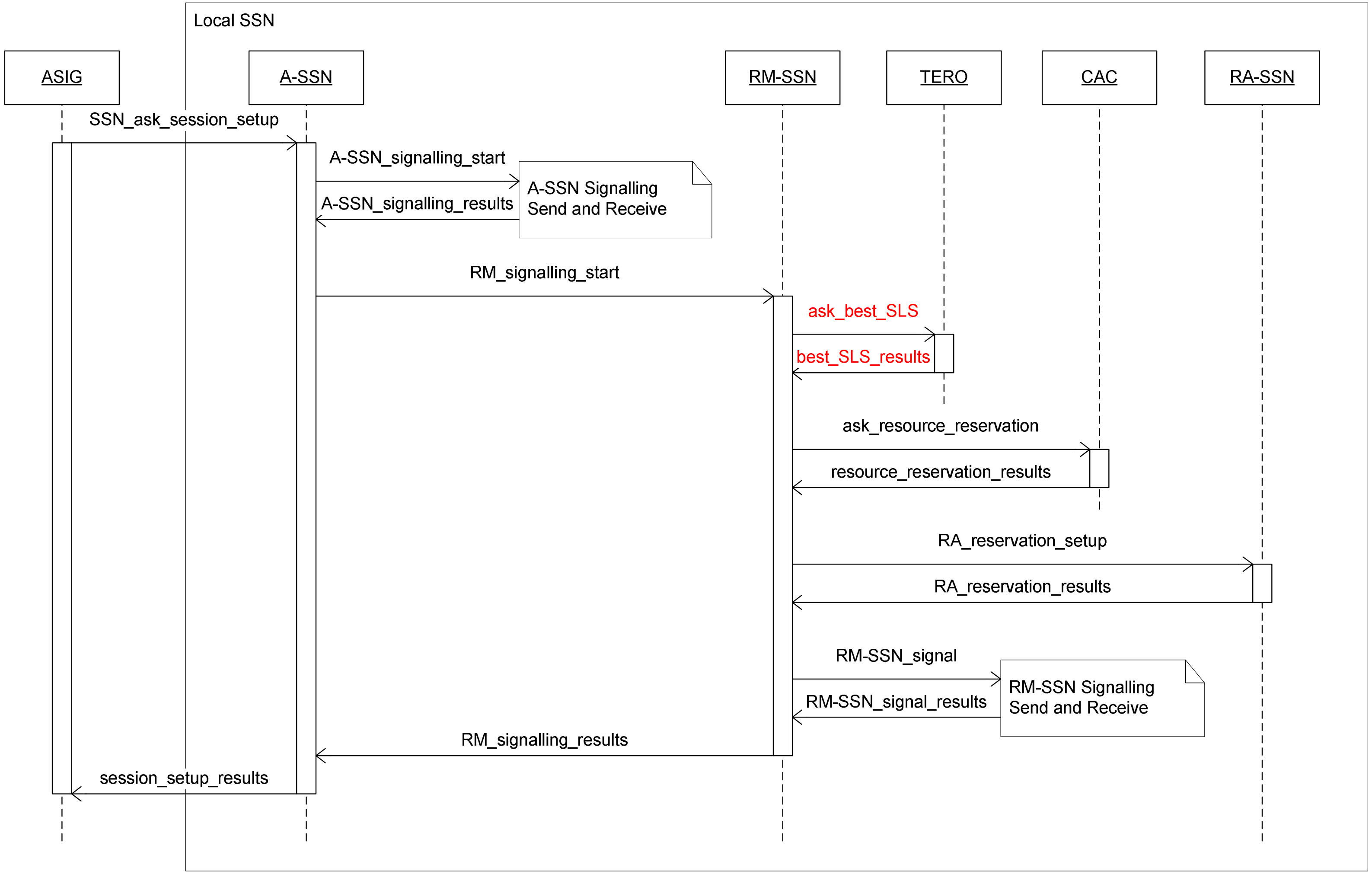


Figure 41: Sequence diagram of session setup by ASIG

9.3.2 Session update by ASIG

Once end-to-end communication is established, parameters can be renegotiated or modified. As can be seen in next picture, the diagram is very similar to session setup (**Figure 41**), since any change needs confirmation from each of the TERO, CAC, and RA-SSN modules. In this case the signal RM_signalling_start is substituted by the RM_signalling_update and the RA_reservation_setup by the RA_reservation_update. A session update still involves the TERO and CAC module since it has to verify that resources are available.

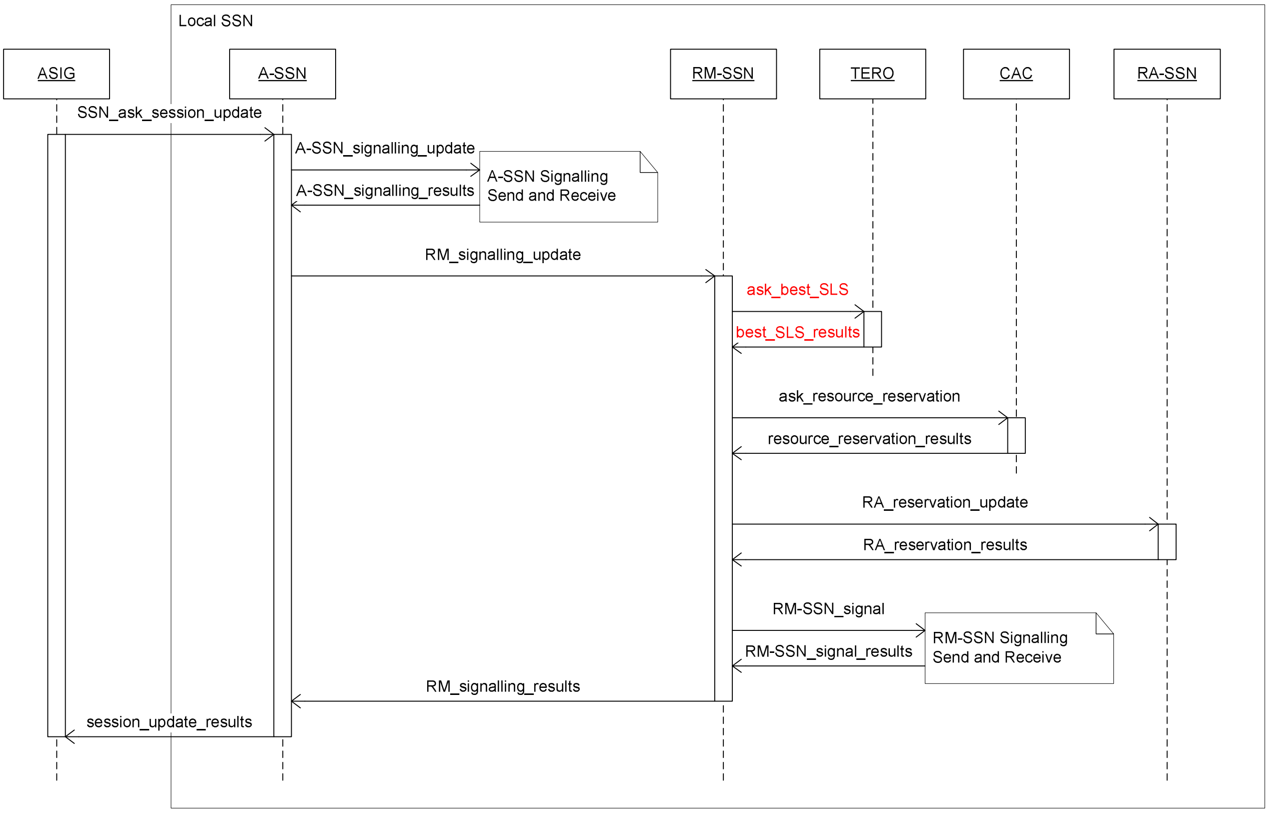


Figure 42: Sequence diagram of session update by ASIG

9.3.3 Session release by ASIG

TERO interaction is not needed in the session release phase. The RM informs the CAC module about the will to release the session so that CAC can update its data.

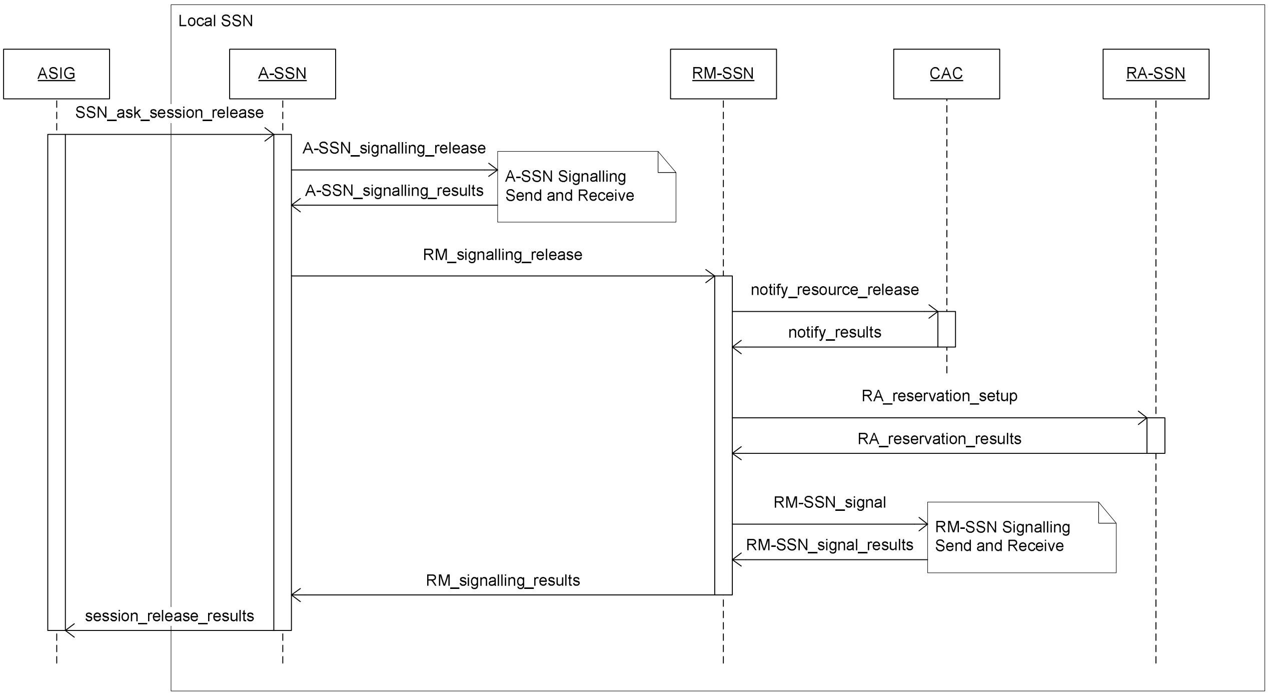


Figure 43: Sequence diagram of session release by ASIG

9.4 Accounting

9.4.1 General description

In this scenario:

1. Each access network at the edges of the communication path is associated with a Local SAAA server, which must generate an accounting record storing the local resources used during the EuQoS session. (In this case, both the Local access network and the Home network are the same since it is a non-roaming scenario)

2. The Local SAAA server for the **calling party** should also keep an accounting record for the **global** session parameters.

3. **All remaining SAAA** systems along the resource allocation path must collect accounting information about **local** resource reservations for each

EuQoS session. Thus, every network provider involved in the EuQoS session will have detailed information that will/could be used to charge the Home network for the resource usage. Thus all actors in the EuQoS path can charge for this service.

4. When the communication concludes, all accounting records will be sent to the corresponding local charging system.

5. Finally, there are two different options:

    i. The billing module at calling party, i.e. the Home CHAR system if a classical telephony model is followed, will collect all charging information from different Local CHAR systems. It will create a bill for the end user and it will pay each of the different Network/Service Providers involved during the EuQoS session. (**Note**: It is necessary to develop a charging plan because some questions are not answered yet: Who pays for a bi-directional connection? What if the QoS is different in each direction of an interactive session?)

    ii. A clearinghouse could collect all charging information from different Local CHAR systems. This entity would act as a common trusted entity. It will be in charge of handling the service level agreements and billing processes among different EuQoS Network/Service Providers. It will create the bill for the end user and it could forward it to the end user transparently on behalf of its EuQoS Service Provider.

9.4.2 Detailed process

Once the session has been granted, the accounting should start. None of the network providers will permit an unpaid use of their resources. Hence, they are willing to loose a little in return for better efficiency. Operators keep a track of everything that happens in their network.

The A-SSN is in charge of notifying each SAAA server through the resource reservation path to start the accounting procedure by means of ”SAAA_notify_acc_event (requestType: START)”. This signal will be sent by the A-SSN.

However, there are two possibilities:

1. Send this signal only when A-SSN receives the “200 OK” SIP-message confirming that the path was successfully established. This solution would not be fair from the user’s point of view since if the callee does not pick up the call, the communication never begins, but the calling user is billed. However, all the network providers (including the core network and access network) already have

resources reserved and, therefore, they might want to charge for this resource allocation (even though the resources were **not** actually used).

2. Send this signal after the ringing tone stops, i.e. once when the callee accepts the incoming traffic stream. It is the "fairest" for a user. Now the user will only be charged when the SIP clients actually send data to the other end.

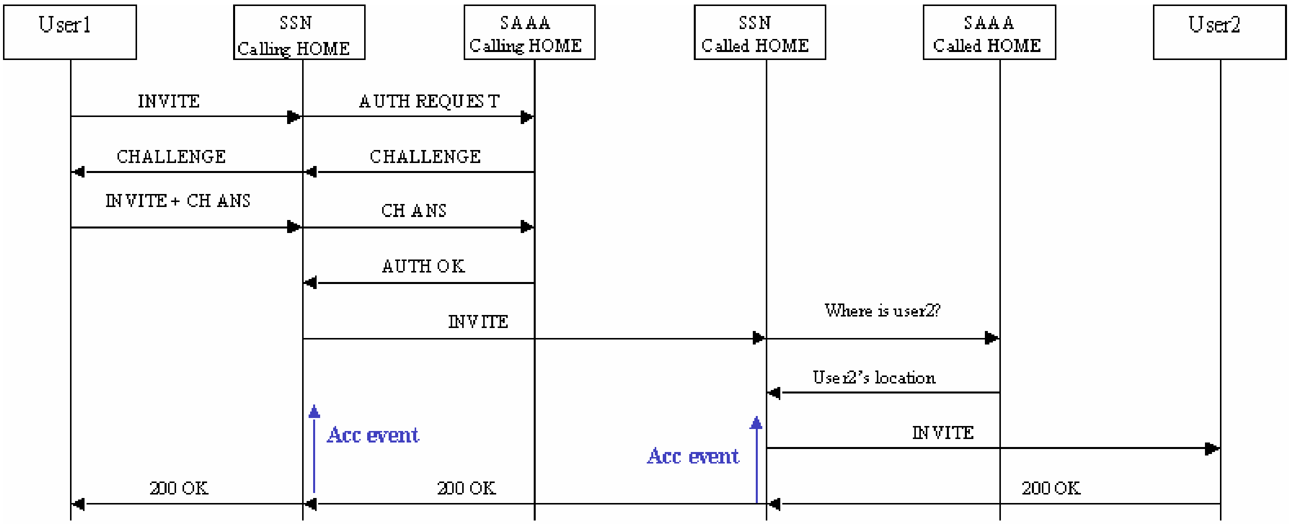


Figure 44: Non-roaming accounting events

From this moment on, every SAAA server records information about the session that the EuQoS system is providing. **Each** SAAA server will be informed if any parameter is changed during the communication via a SAAA_notify_acc_event (requestType: INTERIM), and each server will also be notified when the session ends (whenever the "SIP BYE" is received) to finish the accounting based upon a SAAA_notify_acc_event (requestType: STOP).

9.4.3 A-SSN accounting data provided to the SAAA.

Figure 45 presents a sequence diagram showing the notification of accounting events sent to the SAAA by the A-SSN. The structure of an AccData object is shown in Figure 46.

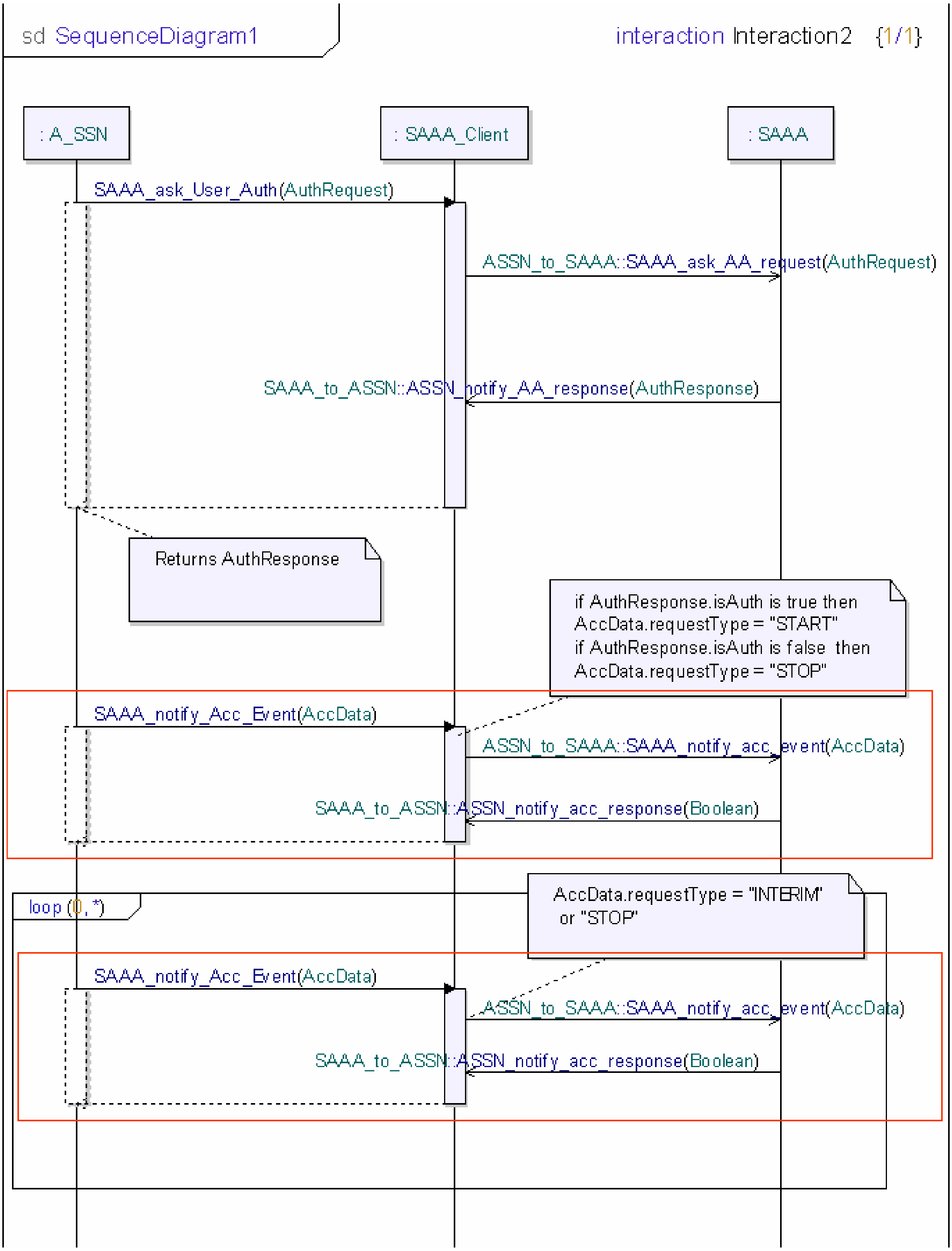

sd SequenceDiagram1
interaction Interaction2 {1/1}
: A_SSN
: SAAA_Client
: SAAA
SAAA_ask_User_Auth(AuthRequest)
ASSN_to_SAAA::SAAA_ask_AA_request(AuthRequest)
SAAA_to_ASSN::ASSN_notify_AA_response(AuthResponse)
Returns AuthResponse
if AuthResponse.isAuth is true then
AccData.requestType = "START"
if AuthResponse.isAuth is false then
AccData.requestType = "STOP"
SAAA_notify_Acc_Event(AccData)
ASSN_to_SAAA::SAAA_notify_acc_event(AccData)
SAAA_to_ASSN::ASSN_notify_acc_response(Boolean)
loop (0,*)
AccData.requestType = "INTERIM"
or "STOP"
SAAA_notify_Acc_Event(AccData)
ASSN_to_SAAA::SAAA_notify_acc_event(AccData)
SAAA_to_ASSN::ASSN_notify_acc_response(Boolean)

Figure 45: Notification of accounting events to SAAA

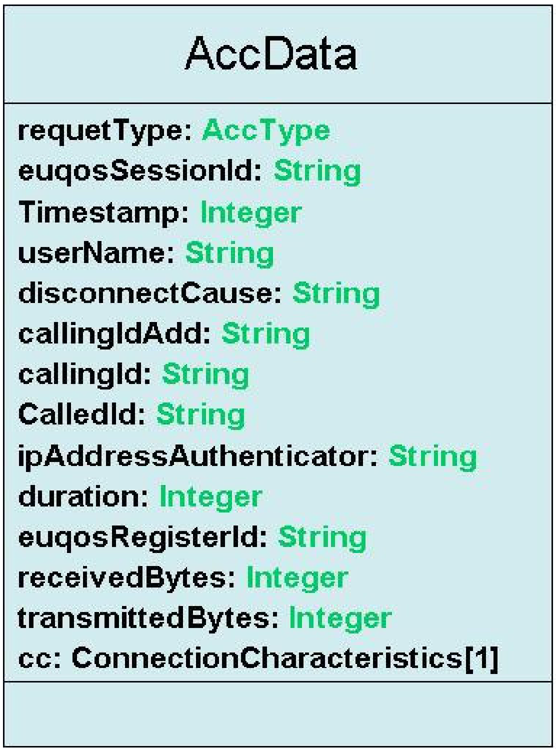


Figure 46: Accounting Data structure

The AccData object is composed of:

- *requestType:* START, STOP, INTERIM
- *euqosSessionId*: uniquely identifies the session in the EuQoS system
- *timestamp:* is the time when the accounting data packet was received
- *username:* the EuQoS username of the user that started the session
- *disconnectCause:* the cause of the session disconnection. If the *requestType* is START or INTERIM this value is not meaningful.
- *callingIpAdd:* the IP address of the user that originated the request message
- *callingId:* the user that originated the request
- *calledId:* the user that replies to the session
- *ipAddressAuthenticator:* the A-SSN's IP address
- *duration:* the duration of the session. If *requestType* is START this value is set to 0. If *requestType* is INTERIM or STOP this value is the difference from the START timestamp and the current timestamp.
- *euqosRegisterID:* the EuQoS identifier that uniquely identifies the user subscription.

- *receivedBytes:* the number of bytes received from the START timestamp up to the current timestamp. If *requestType* is START this value is set to 0.

- *transmittedBytes:* the number of bytes transmitted from the START timestamp up to the current timestamp. If *requestType* is START this value is set to 0.

- *ConnectionCharacteristics, MediaDescription, QoSCharacteristics, and Codec* objects are the same as those shown before in 9.2.3.

The information conveyed by AccData over the DIAMETER protocol is the same as what the SAAA will forward to the charging system (CHAR) via the Java Message Service[4] (JMS). In order to simplify the translation process an XML Schema[5] has been defined for AccData. This schema contains public representations of other object data types like ConnectionCharacteristics, MediaDescription, CODEC, and QoSCharacteristics described previously in this document. The following picture depicts the XML Schema for AccData.

[4] The Java Message Service (JMS) API is a messaging standard that allows application components based on the Java 2 Platform, Enterprise Edition (J2EE) to create, send, receive, and read messages. It enables distributed communication that is loosely coupled, reliable, and asynchronous. More information can be found in http://java.sun.com/products/jms/overview.html

[5] The Extensible Markup Language (XML) is a World Wide Web Consortium-recommended general-purpose markup language for creating special-purpose markup languages, capable of describing many different kinds of data. Its primary purpose is to facilitate the sharing of data across different systems, particularly systems connected via the Internet. (http://en.wikipedia.org/wiki/XML)

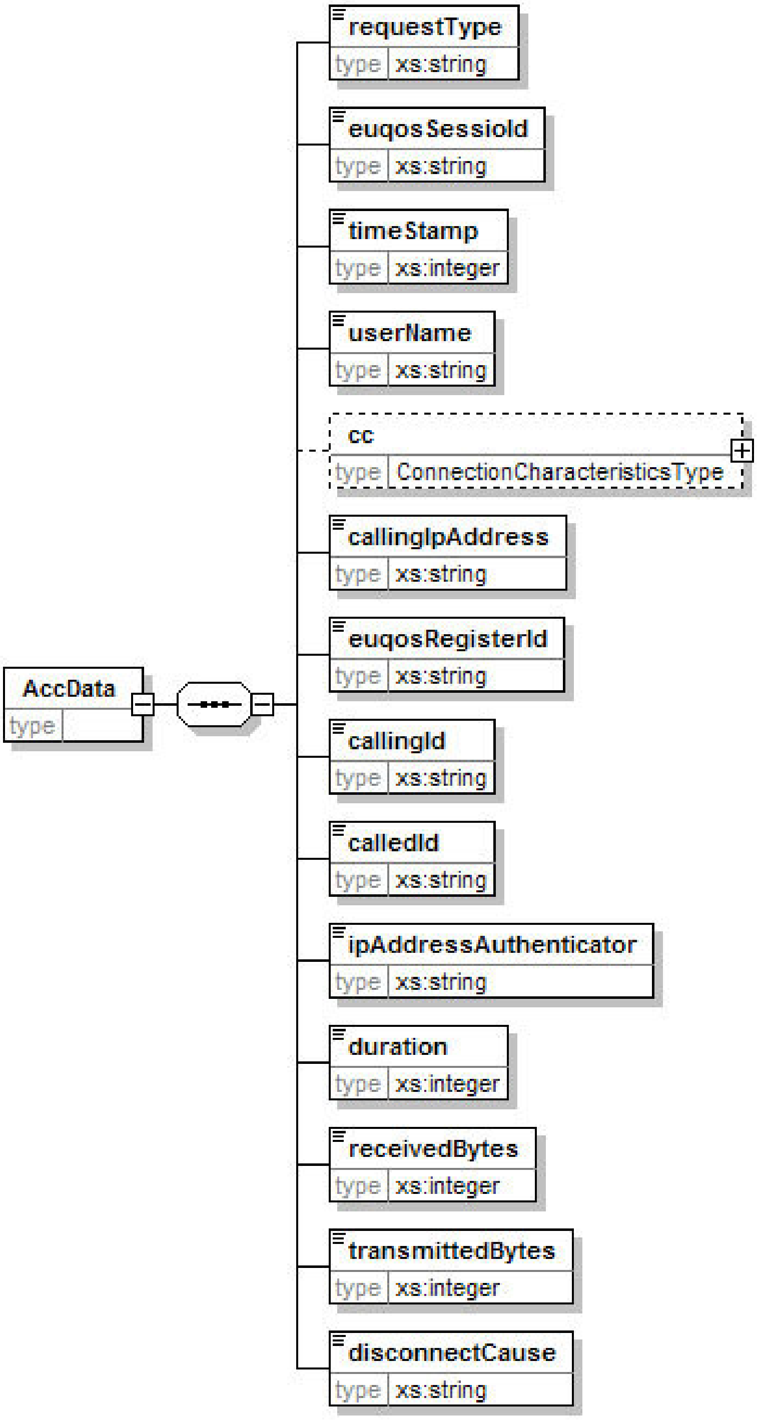


Figure 47: AccData XML Schema

9.4.3.1 EuQoS IPDR structure

The IPDR data structure shall contain all the QoS parameters exchanged among software modules and it shall drive this information from SAAA to CHAR software module.

The following XLM code shows an example of an IPDR structure.

```
  <?xml version="1.0" encoding="UTF-8" ?>
- <IPDRDoc docId="f7da7b2f-0102-5000-f577-ad050a2804dd" creationTime="2005-03-
     31T09:04:13.083Z" IPDRRecorderInfo="EuQosIPDRDocTest" version="3.1"
     xmlns="http://www.ipdr.org/namespaces/ipdr"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="http://www.ipdr.org/namespaces/ipdr EuQoS.xsd">
- <IPDR xsi:type="IPDR-EuQos-Type">
<IPDRCreationTime>2006-03-08T13:20:00Z</IPDRCreationTime>
<seqNum>1234</seqNum>
<accountingRecordType>Start</accountingRecordType>
<EuQoSSessionID>1111111111</EuQoSSessionID>
<userName>Jesús</userName>
<hostname>Piton</hostname>
<callingIPAddress>10.150.0.2</callingIPAddress>
<calledIPAddress>10.150.0.3</calledIPAddress>
<callingID>calling@euqos.org</callingID>
<calledID>called@euqos.org</calledID>
<applicationType>VoIP</applicationType>
<authenticatorIPAddress>10.150.0.1</authenticatorIPAddress>
<startTime>20065-03-08T13:20:00Z</startTime>
<endTime>2006-03-08T17:20:00Z</endTime>
<timestamp>2006-03-08T17:20:30Z</timestamp>
<sentBytes>10000</sentBytes>
<receivedBytes>10001</receivedBytes>
<duration>60</duration>
<disconnectReason>Normal</disconnectReason>
<QoSRequested.profileName>Gold</QoSRequested.profileName>
<QoSRequested.maxGuaranteedBw>200</QoSRequested.maxGuaranteedBw>
<QoSRequested.maxDelay>50</QoSRequested.maxDelay>
<QoSRequested.maxJitter>0</QoSRequested.maxJitter>
<QoSRequested.networkService>BE</QoSRequested.networkService>
<QoSDelivered.profileName>Gold</QoSDelivered.profileName>
<QoSDelivered.maxGuaranteedBw>100</QoSDelivered.maxGuaranteedBw>
<QoSDelivered.maxDelay>40</QoSDelivered.maxDelay>
<QoSDelivered.maxJitter>0</QoSDelivered.maxJitter>
<QoSDelivered.networkService>BE</QoSDelivered.networkService>
  </IPDR>
<IPDRDoc.End count="1" endTime="2006-03-31T09:04:13.143Z" />
  </IPDRDoc>
```

9.4.4 CHAR interface

The taxonomy used in the API exposed by the CHAR module is:

<return_results> <modulename>_<notify|ask|get|set>_<function_name>(<list of parameters>)

In which:

<return_results> the values returned from a interface call;

<modulename> the name of the module that exposes the interface;

<notify|ask|get|set> the action requested to the module;

<function_name> the name given to an exposed interface;

<list of parameters> the list of parameters given to interface call.

CHAR Interface	**Activated By**	**Interface Description**
notify_result **CHAR_notify_charging_events** (charging_events_parameters)	**SAAA**	This interface provide to the SAAA module the way to notify (CHAR module) new events about the user's sessions
user_account_info **CHAR_get_account_info** (retrive_request)	**Web Client**	This interface provide to the end user the way to access to his account information (invoice, billing, personal info, etc) Merge into one
user_account_configuration **CHAR_set_account_configuration** (account_configuration)	**Web Client**	This interface provide to the end user the way to modify his account configuration (QoS levels, personal info, etc) merge into one

Table 2: CHAR interface

10 Use case 2: Roaming scenario

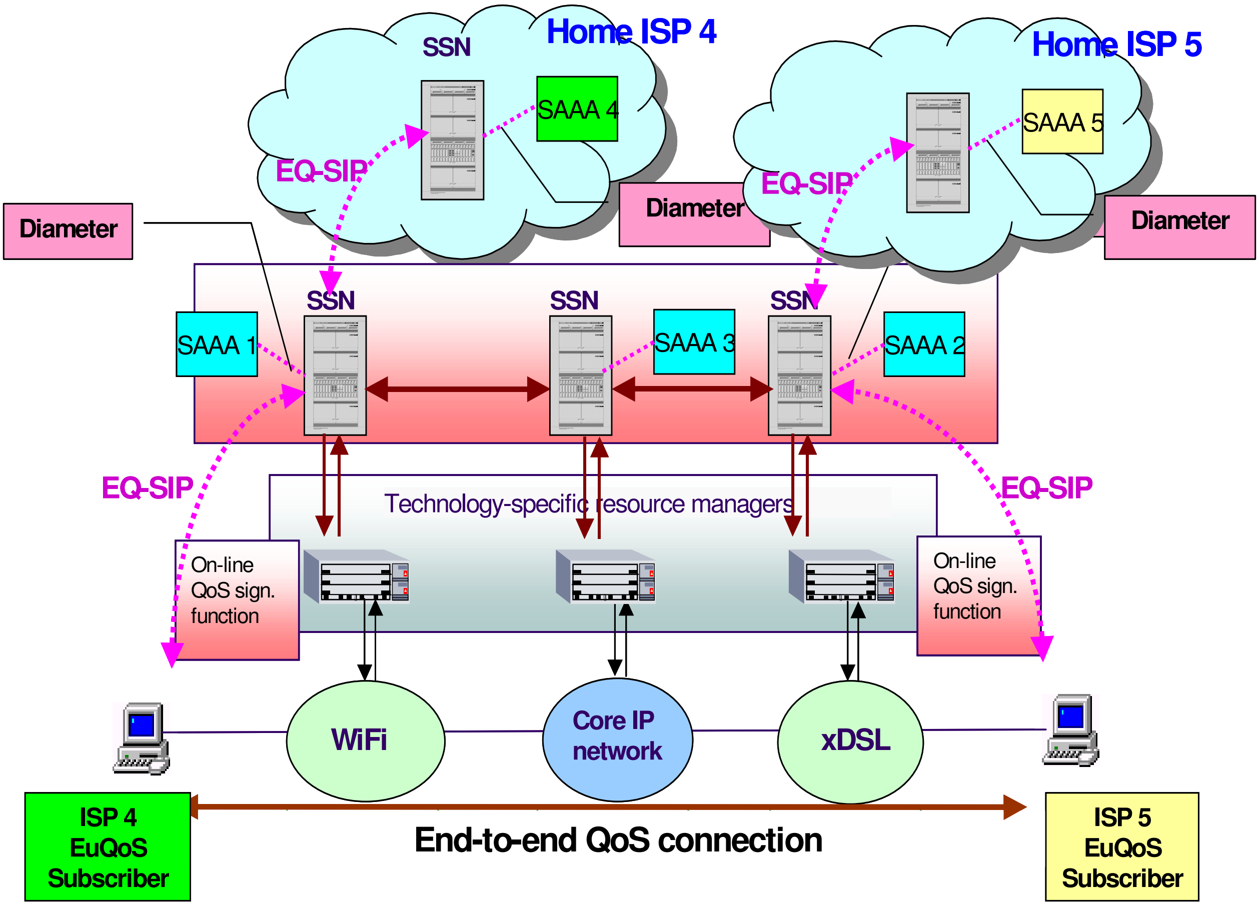


Figure 48: Roaming diagram

In this scenario the calling party is temporally visiting ISP1's (Internet Service Provider) network. It is registered in its home network (in ISP4's domain), but EuQoS service needs to be offered by ISP1, if the user is to receive a specific QoS. The calling party tries to communicate with another user, the called party, currently connected to ISP2, but who is subscribed to ISP5's services. In this scenario it is assumed that **all** ISPs support EuQoS. The non-EuQoS scenario will be mentioned in section 11.

10.1 Provision of EuQos service

The provisioning of the service is performed in the same way as in the non-roaming case. However, one more field should be added to the data structure provided to the SAAA in order to inform the SAAA when/where from the user will be able to roam. Since the Auth process is **always** at the Home network, it is only the Home network where you need to know if the user is able to roam.

10.2 User Registration: Authentication/Authorization

Before the calling party can use any EuQoS services, the user has to be authenticated and authorized. This authentication/authorization will not occur in the Visited network, but is performed by the **Home** network. When the calling party roams, it connects to an A-SSN node, which is part of the ISP1's network (the Visited network). The calling party needs to contact its Home network for the authentication/authorization, therefore, ISP1 needs to find the exact location of ISP4 (Home) to forward the request.

The first idea to deal with roaming is illustrated below:

SDL Diagram about the Authentication/Authorization in case of roaming access

The SDL Diagram below shows the communication paths inside and outside the SAAA module in the optional case of roaming access. This diagram proposes Diameter to establish a direct communication between the local Authentication Server and the Home Authentication Server.

The details regarding Accounting and Registration have been omitted (see the previous section for more information about these).

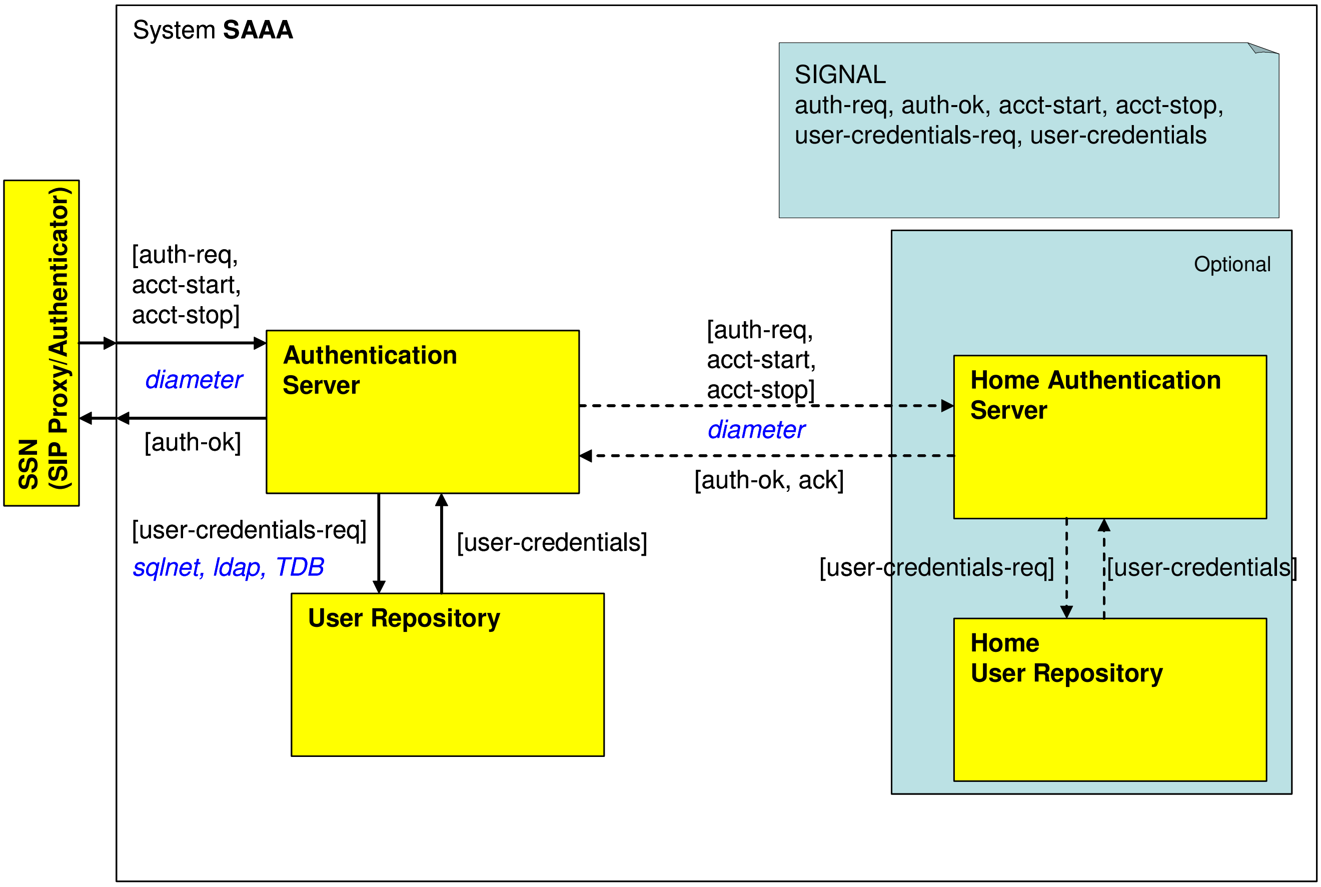


Figure 49: SDL diagram about Authentication/Authorization in case of roaming access

UML Sequence Diagram

The UML Sequence Diagram describes the message sequence between the entities:

- UA-Supplicant (SIP User Agent acting as a Supplicant at application level);
- SIP Proxy-Authenticator (SIP Proxy Server acting as an Authenticator inside SIG Function);
- Authentication-Server (a DIAMETER or RADIUS Authentication Server inside SAAA Function);
- A Remote Authentication-Server (a DIAMETER or RADIUS Remote Authentication Server) in order to show the system behavior in the optional case of roaming access.

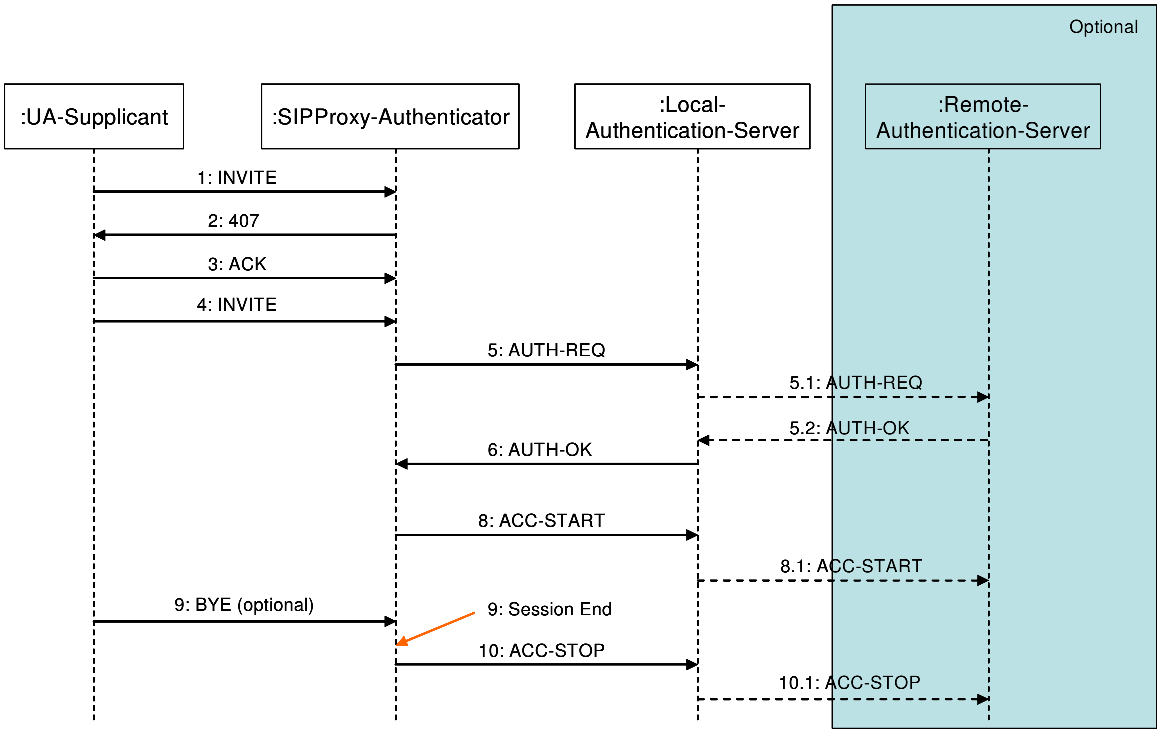


Figure 50: SAAA UML diagram

However, this method has an enormous disadvantage as all SAAA systems would need to have a complete network topology repository in order to contact other SAAAs. This makes the solution non-scalable.

Alternatively, the Local Authentication Server could simply redirect the request to the user's remote Home domain authentication server. This solution is not realistic. Nowadays, a direct communication between SAAA servers from different operators is simply non-realistic. The SAAA contains critical information about users and services. Furthermore, a new connection per roaming AS represents a huge cost in terms of links and security, which translates to more money. Thus, it is not advisable to interact directly with any other network. Besides, this method implies an enormous disadvantage. All SAAA systems would therefore need to have a complete network topology repository in order to contact another SAAA. This makes the solution non-scalable.

Therefore, two more efficient approaches can be considered:

10.2.1 SAAA Broker approach

In this case, the authentication/authorization request will be forwarded from the local SAAA server to a known SAAA Broker. This entity keeps an updated schema of the topology of the whole network and information about the SLAs between different EuQoS providers. The broker has trust relationships to each of the different providers; i.e. each EuQos provider has a trust relationship with the broker and thus the broker can act as a Certification Authority to other EuQoS providers. The broker will send to the Visited SAAA server the address of SAAA server at ISP4 (i.e., the user's Home network) or the broker could also redirect the request directly to the real destination by means of DNS. The broker can apply policies concerning the roaming subscribers. This new entity could be a new module in an existing SAAA (co-located configuration) or be an external component (stand-alone configuration).

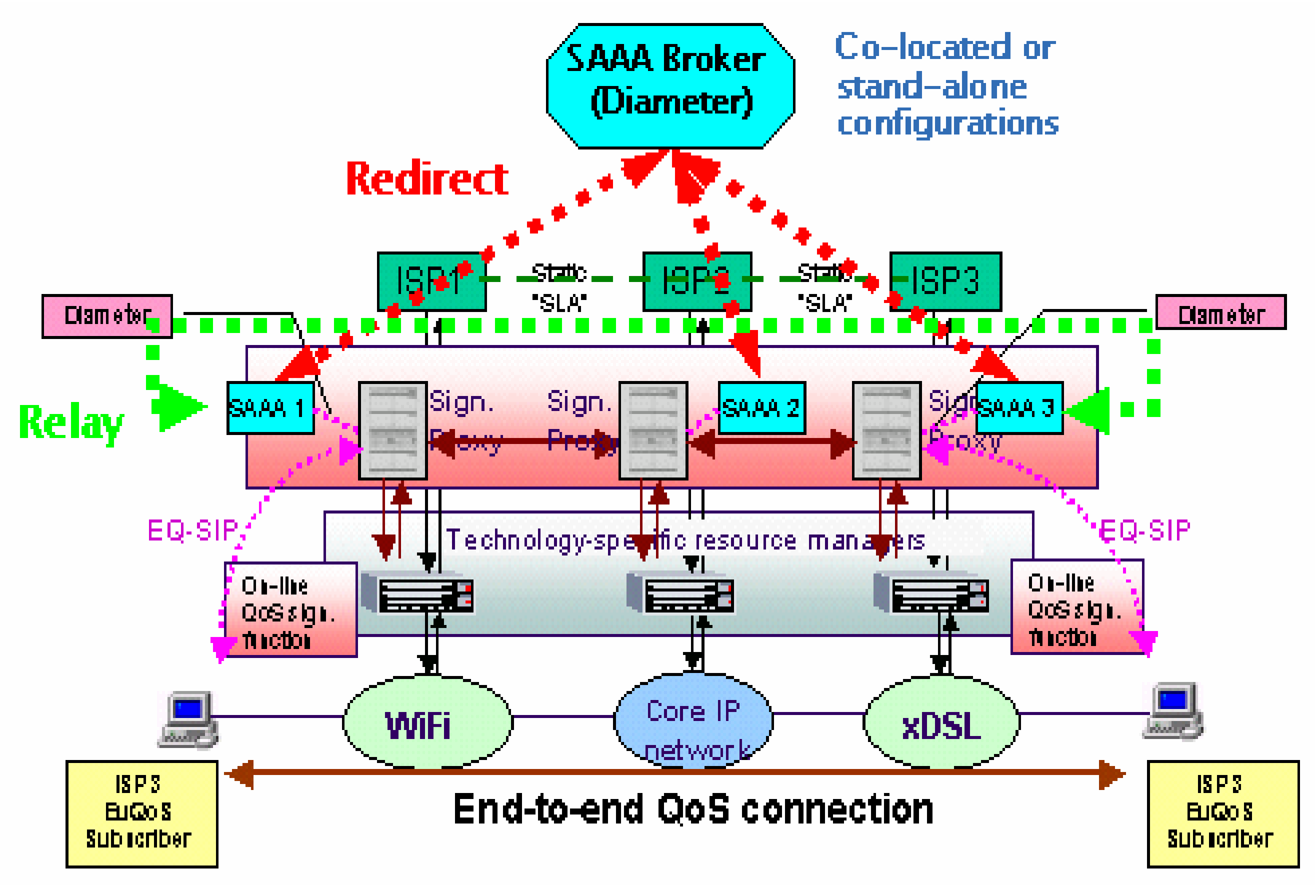


Figure 51: SAAA Broker

10.2.1.1 Diameter Redirect

A broker model allows the end Diameter servers to directly communicate (Figure 52). In this model the broker simply provides redirect services, which is aimed at reducing the amount of configuration that would otherwise be necessary on all end Diameter servers. When a Diameter server sends a request to the broker, the broker returns contact information that is then used by the requesting server to re-issue the request directly to the home Diameter server. In order for the end Diameter servers to be able to communicate in a secure fashion, a pre-established security association is required. This can be in the form of a long-lived shared secret, which has scalability problems, or via certificates when the broker's organization provides Certificate Authority services. In the event that the broker also provides settlement services, it is possible for the accounting information, signed by both parties, to be transmitted to the broker by the server providing service to the user.

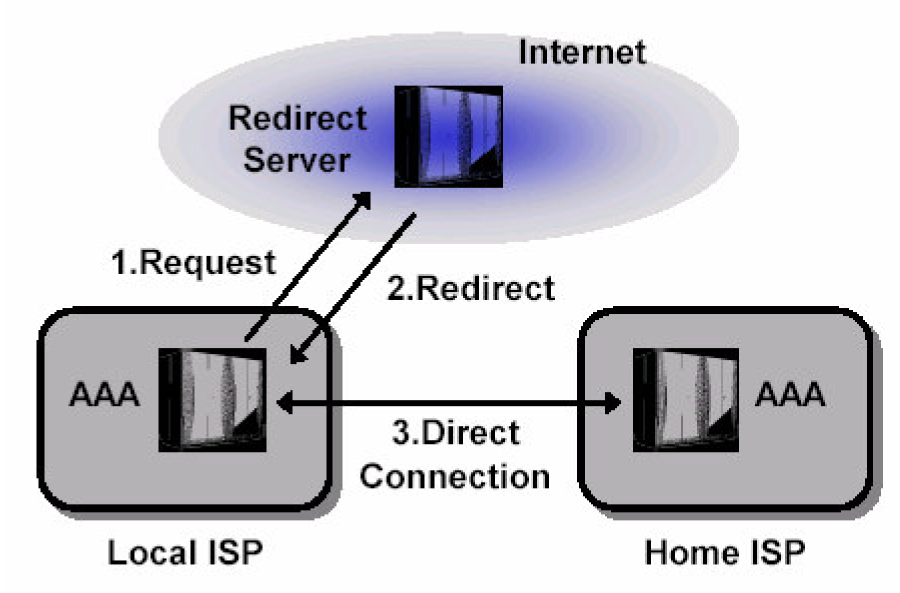


Figure 52: Redirect illustration

When the broker provides the message forwarding functions, it can validate that the source and destination DIAMETER servers are in "good standing", which reduces the processing on the end servers. This can be done by having the broker check the server's certificates, via an online certificate status protocol [70], or through local configuration. The broker can **optionally** attach the source server's certificate if it isn't already present in the message. When a broker receives a request from or destined to a realm that is either unrecognized or no longer part of the roaming consortium, an error will be returned to the requesting server.

In cases of inter-consortium roaming, the brokers involved can be responsible for validating any certificates involved. Note that it is also possible for the broker to periodically issue new certificates to the roaming consortium members out-of-band in order to eliminate the need to add certificates to each message, decreasing the message size and the per-message processing penalty.

When the broker provides redirect services, the broker can return both the source and the destination server's certificates. The certificates are encapsulated within a DIAMETER attribute, and include a timestamp, an expiration time all signed by the broker.

The end server must be setup such that they will trust the certificate returned by the broker, they do not have to perform any additional certificate validation checks. However, local policy may require that the end DIAMETER servers validate with the broker periodically.

Note that even though some broker's do allow direct communication, some will require that all accounting messages be forwarded by the broker. This is typically required when the broker also provides settlement services. In such a network, the broker normally requires some reassurances that the user was in fact authenticated and authorized by the home DIAMETER server prior to accepting accounting records. This can be achieved by requiring that both DIAMETER servers sign the Accounting data in a serial fashion [71].

10.2.2 SIP Proxying

The authentication/authorization request is forwarded through the signaling path using EQ-SIP in the SIP Proxying approach. This technique avoids the need for the broker module, but adds some complexity to SSN nodes. The originating SSN will send the request to the Home SAAA server through the SSN SIP proxy network. Subsequently, the Home SAAA will generate an answer and send it back to the Visited network: i.e. to the SSN and roaming subscriber.

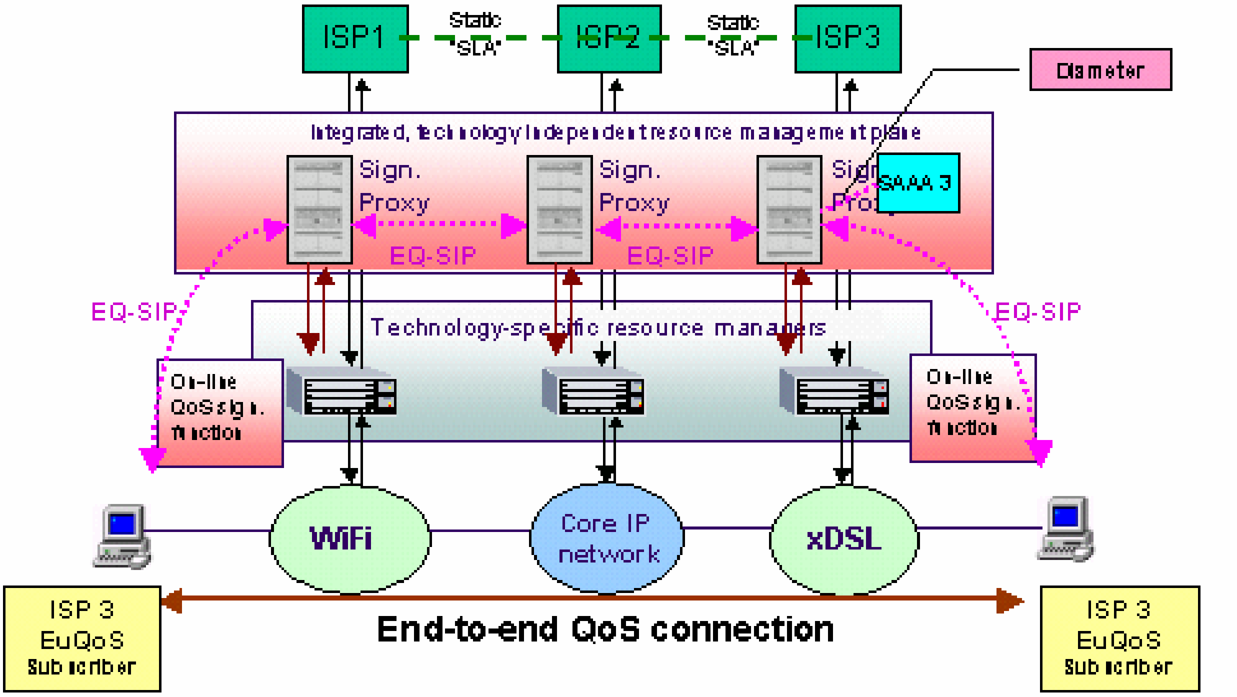

ISP1
Static SLA
ISP2
Static SLA
ISP3
Diameter
Integrated, technology independent resource management plane
Sign. Proxy
Sign. Proxy
Sign. Proxy
SAAA 3
EQ-SIP
EQ-SIP
EQ-SIP
EQ-SIP
Technology-specific resource managers
On-line QoS sign. function
On-line QoS sign. function
WiFi
Core IP network
xDSL
ISP 3 EuQoS Subscriber
End-to-end QoS connection
ISP 3 EuQoS Subscriber

Figure 53: SIP Proxy

A lack of availability of the service to an EuQoS authorized user in the Visited network is possible in this scheme. It would take place if the Home network had authorized the user, but there were no roaming agreement between the Visited and Home ISP. In such a case, the communication will be dropped off since the Visited network does not allow this user to roam. Therefore, it is necessary to check the user's realm at the SSN (SIP proxy server) before forwarding the authentication/authorization request to avoid unnecessary traffic.

Roaming Restriction

A "Roaming Restriction" feature could also be applied at Home SAAA:

When an authentication/authorization message is received and a roaming restriction function is active, then the calling SSN's Fully Qualified Domain Name (FQDN) is analyzed and if it does not belong to the home network:

- If the user is not allowed to roam outside the home network, then the operation is not accepted and DIAMETER_AUTHORIZATION_REJECTED error is returned.
- If there is not a roaming agreement with the visited network, then the operation is not accepted and DIAMETER_ROAMING_NOT_ALLOWED error is returned.

Note: if the FQDN belongs to the home network, then it is not a roaming case. Hence no Roaming Restriction will be applied.

10.2.3 User Registration

When a user is not within their home domain, then the REGISTER request will be forwarded from the Visiting network to the Home network.

Example:

From:	<sip: user@euqos1.org>
To:	<sip: user@euqos1.org>
Contact:	<sip: user.euqos@euqoshome1.org>

The following figure illustrates how the REGISTER procedure is performed.

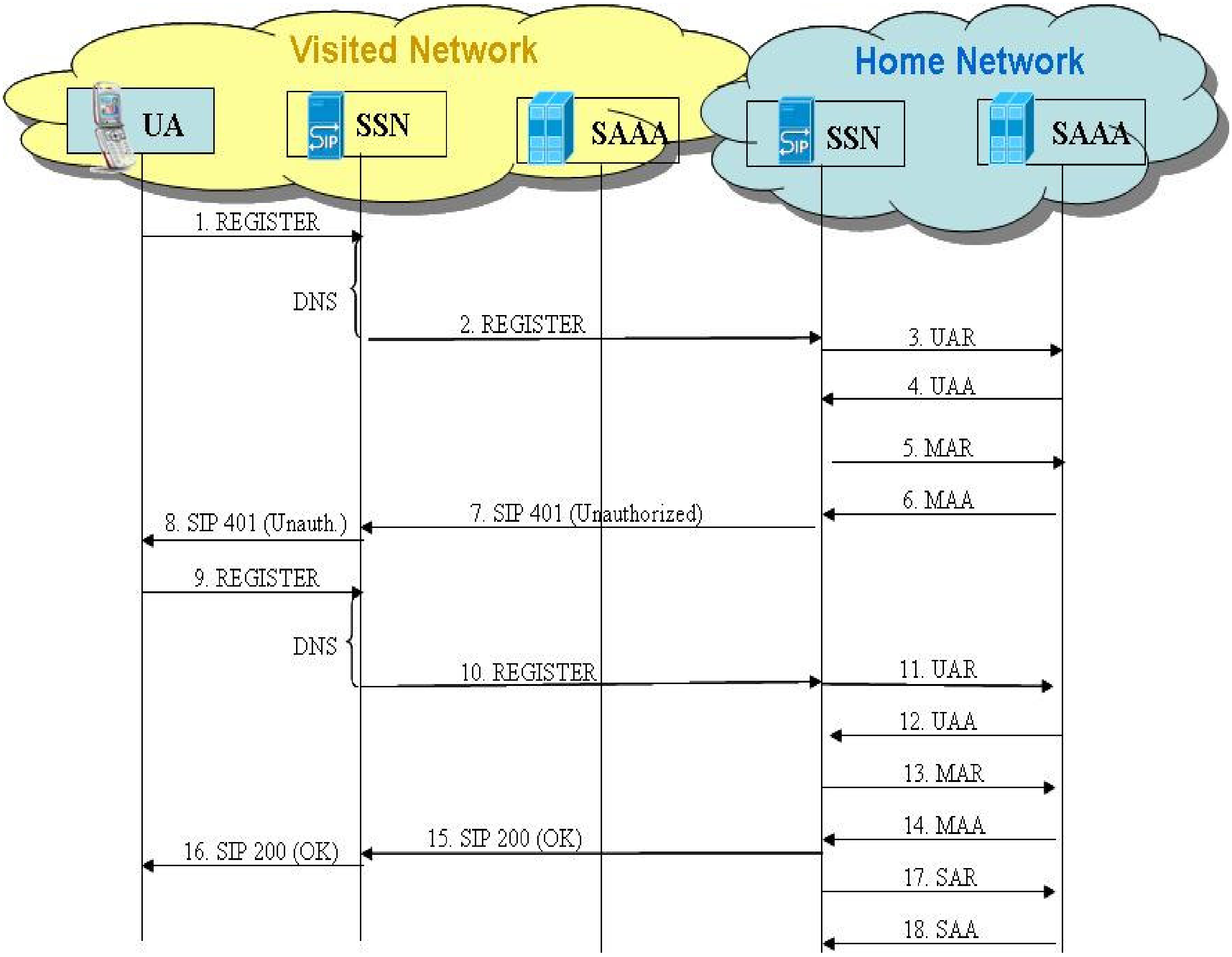


Figure 54: Roaming user registration

The user agent performs the REGISTER request (1). The local SIP Proxy is able to determine if the user is roaming or not, therefore, the local A-SSN checks the user's permissions and compares the user's realm with its own realm. If the realm of the calling party is different than the local network realm, then the user is roaming. The following step will be to check if there are established agreements with the Home network for any roaming service. Otherwise, the REGISTER request will be rejected.

In a roaming case, the local A-SSN (SIP Proxy) forwards the REGISTER request to the user's Home Network. If the SIP address that identifies the user is within an Address Of Record (AOR) scheme, the local SIP server performs a DNS query to locate the SIP server in the home network. The look up in the DNS is based on the Home SIP server's URI. The DNS provides the visiting SIP server (A-SSN) with an address of the A-SSN in the home network. Otherwise, i.e., if the SIP address uses the Fully Qualified Domain Name (FQDN) or the IP address (identifies a device) of the host, which needs no resolution for routing.

Then, the local SIP server forwards the SIP REGISTER request (2) to the Home SIP server. The DIAMETER client in the calling Home SIP server contacts its Diameter server by sending a Diameter User-Authorization-Request (UAR) (3) to determine if this user is allowed to receive service, and if so, request the address of a local SIP server capable of handling this user. The Diameter server answers with a Diameter User-Authorization-Answer (UAA) message (4), which indicates a list of capabilities that the receiving SIP server (SIP server 1) may use to select an appropriate SIP server (SIP server 2) and/or a SIP or SIPS URI pointing to SIP server 2.

After this, the AAA server the Diameter client in the Home SIP server requests user authentication from the Diameter server by sending a Diameter Multimedia-Auth-Request (MAR) message (5). The Diameter server responds with a Multimedia-Auth-Answer (MAA) message (6) with the Result-Code AVP set to the value DIAMETER_MULTI-ROUND-AUTH, and also generates a nonce that includes a challenge in the MAA message. The Home's SIP server uses that challenge to map into the WWW-Authenticate header in the SIP 401 (Unauthorized) response (7), which is sent back to the SIP local server and then to the SIP UA (8).

The local SIP server receives a next SIP REGISTER request containing the user credentials (9). Note that the SIP server does not need to keep a state, and there is no guarantee that the SIP request arrives at the same SIP server; i.e., there could be a farm of SIP servers operating in redundant configuration. Therefore, it could also be necessary to perform a DNS query to resolve the SSN Home's SIP address. Once this address is known, the local SIP server forwards the SIP REGISTER request to the address of the Home SIP server (10).

Steps (11) and (12) are needed for the same mean explained before in steps (3) and (4). The Home SIP server extracts the credentials from the SIP REGISTER request and its Diameter client sends those credentials in a Diameter MAR message (13) to the Diameter server at the Home AAA. At this point, the Diameter server is able to authenticate the user, and upon success, returns a Diameter MAA message (14) with the AVP Result-Code set to the value DIAMETER_SUCCESS. The Home AAA also stores the current location of the user and the binding between the user's SIP address and its contact address.

After this, the Home SIP server generates a SIP 200 (OK) response (15), which is forwarded to the local SIP server and eventually to the SIP UA (16).

If the Diameter client in the Home SIP server is interested in downloading the user profile information or is required to store the address of the SIP server in the Diameter server, then the Diameter client sends a Diameter SAR message (17) to the Diameter server. The Diameter server replies with a Diameter SAA message (18) that contains the requested **user profile** information and the acknowledgement of the SIP server address storage. These actions are needed when the SIP server has to retrieve a user profile used to provide services to the served user, or when the SIP server keeps a state for the user, so the Diameter server needs to store the SIP server's address.

SIP registrar, by nature, is soft-state; this means that registration bindings must be periodically refreshed (updated). The expiration time of a binding is indicated by the registering entity using the expires parameter in a Contact header. If this parameter is not present, the registrar assumes an expiration time of 1 hour. If the user agent does not refresh or otherwise explicitly remove the binding, the registrar silently removes it when the expiration time lapses.

Anytime the user enters a new domain, a new registration shall be mandatory, so that the Home network updates the current location of the user.

10.2.4 Scalability

The basic architecture assumption of this document is that all the data related to a user is stored in a unique Diameter server. Contrary to general opinion, this does not create a single point of failure. It is assumed that Diameter servers are configured in a redundant fashion in an attempt to mitigate such a single point of failure problem.

In large networks, where the number of users may be significant, there might be a need to scale the number of Diameter servers. All the data associated with a user is still stored in one logical Diameter server (typically operating in a redundant configuration), but the data associated with different users may reside in different Diameter servers.

Although this configuration scales well, it introduces a new problem, namely: given the user's SIP AOR as an input, how to determine which of the Diameter servers stores the data for that particular SIP AOR. This problem can be solved with inspiration from the Diameter redirection mechanism specified before in 10.2.1.1.

A solution for the previously stated problem is the addition in the architecture of a new Diameter node, the Diameter Subscriber Locator (SL). The Diameter SL contains a database or routing tables that map SIP AORs to Diameter server URIs. A particular Diameter server URI points to the actual Diameter server that stores all the data related to a particular SIP AOR, and in consequence, to the user who owns the SIP AOR. The Diameter SL acts in a similar way to a Diameter Redirect Agent, dispatching Diameter requests (e.g., providing the redirection URI in the answer). The Diameter SL can redirect all the request pertaining to a user by setting the Redirect-Host-Usage AVP with a value ALL_USER, as specified in [24].

The Diameter SL can be replicated in different nodes along the network, for the purpose of building scalability and redundancy. The database or routing tables have to be consistent across all these different Diameter SLs, so that equal Diameter requests will produce equal Diameter answers, no matter which Diameter SL processes the request.

10.3 Session setup

A user has to be authenticated every time that he/she wants to access to any service (or operation). When A-SSN receives a new Session_Setup (or INVITE) request, which contains the user credentials previously requested to the ASIG by the A-SSN, the A-SSN needs to ask the SAAA for the authorization to accomplish this request. Therefore the A-SSN initiates an authentication procedure based on a challenge mechanism. This procedure involves the A-SSN, the end-system and the SAAA, as it is shown below.

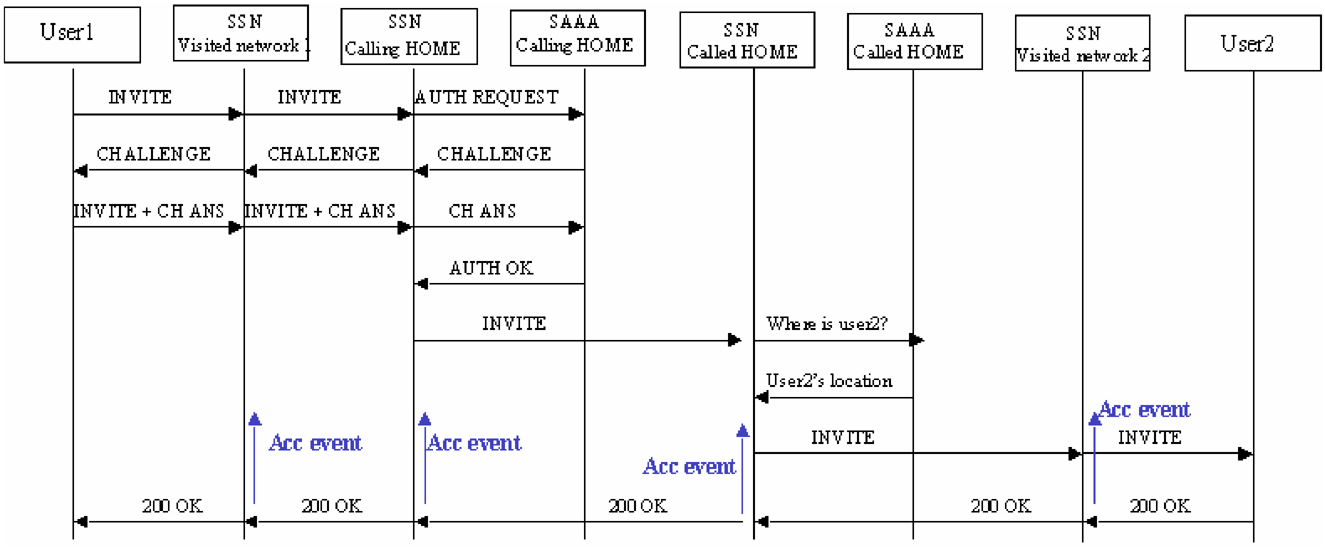


Figure 55: Session Setup high-level diagram

First of all, it is necessary to emphasize that the callee has already been registered in his/her Home network. Otherwise there would be no binding between IP address and public user identity. Therefore, it would not be possible to find the called party and the EuQoS session would never start.

The process begins when the calling party sends the INVITE request. This request is analyzed at the local SSN (SIP Proxy). Looking at the FROM header, the local SIP Proxy is able to determine if the user is roaming or not. In a roaming case, the local A-SSN has to forward the INVITE request to the user's Home Network.

Example:

From:	<sip: user1@euqos1.org>
To:	<sip: user2@euqos2.org>

If the realm of the calling party is different than the local SSN (SIP Proxy) realm, then the user is roaming. The following step will be to check if there are established agreements with the Home network for the roaming service that is being demanded.

In a roaming case, the local SSN (SIP Proxy) has to forward the INVITE request to the user's Home Network. The Home Network will be found based on the user's URI, the SSN determines that the user agent is accessing from a visiting domain and performs a DNS query to locate the A-SSN in the home network. The look up in the DNS is based on the address specified in the Request URI. The DNS provides the visiting A-SSN with an address of the A-SSN in the home network.

Once the INVITE arrives at the Home network, the SAAA home server is asked for the user authentication. If it is successful, then the Home network forwards the request to the called network, euqos2 (i.e. To: <sip:user2@euqoshome2.org>).

The called party could also be roaming. In such a case, the called **Home** network would locate its user with the information provided by its own SAAA (the called party has been already registered). The SAAA stored the current location of the user when the registration was performed. If the user moves from one AS to another, he/she should register again in the new domain.

Since Home's A-SSN is call stateful, it is required to be in the path for all Mobile Originated and Mobile Terminated requests for this user. To ensure this, the A-SSN has to put itself into the path for future requests. One solution of achieving this is to have the A-SSN as the contact point for this user at the home registrar.

After that, the resource reservation takes place. Nothing new has to be considered in this part since it will be solved as in a non-roaming case.

Figure **56** and Figure 57 depict in detail a general scenario where both calling and callee parties are roaming.

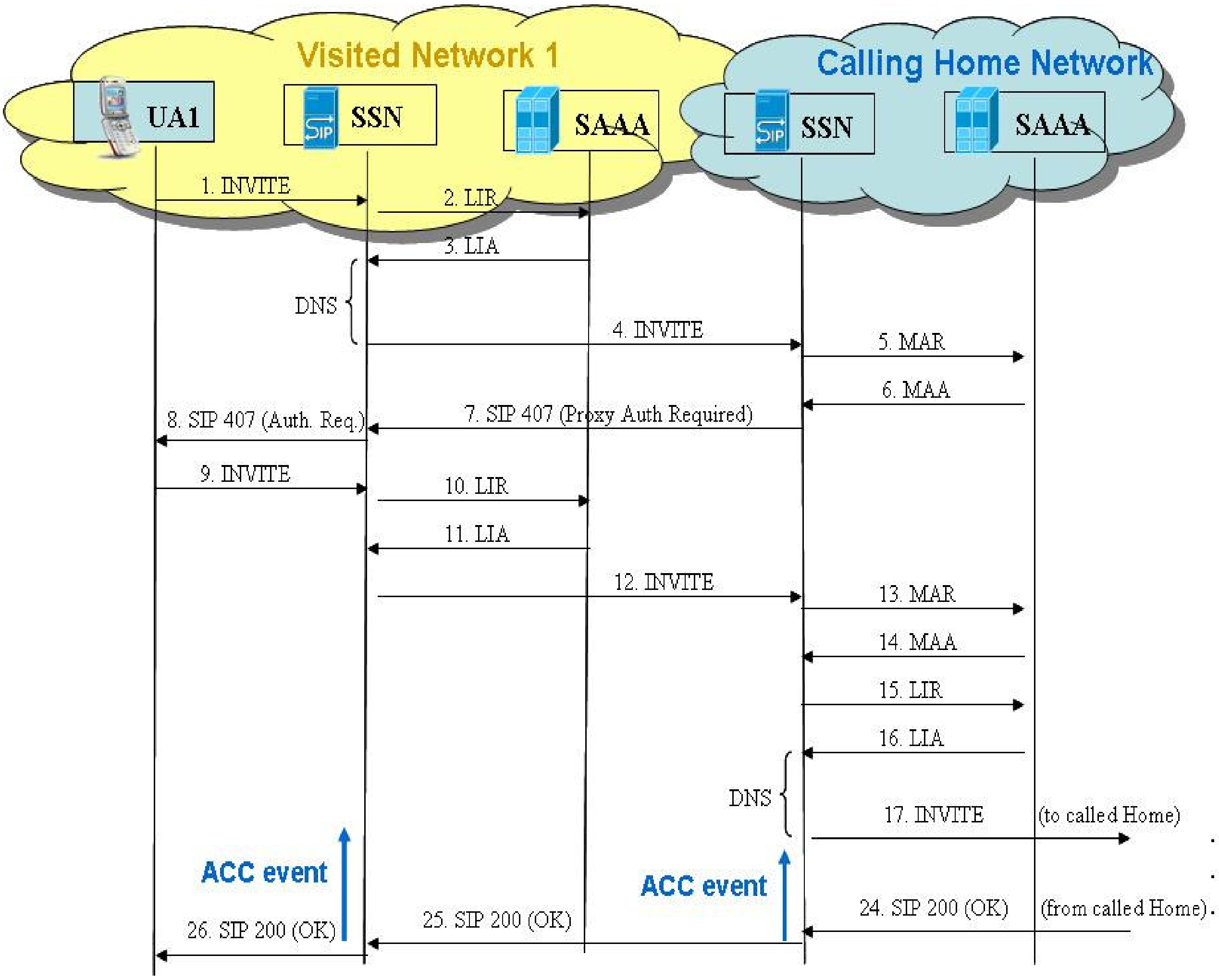


Figure 56: Roaming session setup (1)

According to this figure, a SIP User Agent sends a SIP request to its outbound SIP proxy server (SSN at the visited network) (1). In this case, the message is a SIP INVITE request but it could be any other SIP request. Assuming that this SIP request does not contain any credentials at this time; therefore, the Home SIP proxy server needs to authenticate and authorize the proxy services offered to the user.

The local SIP server needs to find the address of the home SIP server, which is serving the recipient of the SIP request. The Diameter client in the local SIP server sends a Diameter Location-Info-Request (LIR) message (2) to the Diameter server. The Diameter server responds with a Diameter Location-Info-Answer (LIA) message (3) that contains the SIP or SIPS URI of the Home SIP server. The local SIP server forwards the SIP INVITE to the Home SIP server (4), after a DNS query to resolve its address.

Although this example shows the connection between a SIP INVITE request and the Diameter LIR message, any SIP request other than REGISTER (such as SUBSCRIBE, OPTIONS, etc.) would trigger the same Diameter message. (A SIP REGISTER request will trigger a Diameter UAR message).

Then, the Diameter client in the Home SIP server sends a Multimedia-Auth-Request (MAR) message (5). The Diameter server generates a nonce and sends a Multimedia-Auth-Answer (MAA) message (6) that includes the nonce and the rest of the data necessary for the Home SIP server to challenge the user, typically with HTTP Digest Authentication indicated in the MAA message. This data enables the SIP server to create a SIP 407 (Proxy Authentication Required) response (7) that contains a challenge. Once the SIP 407 (Proxy Authentication Required) response message arrives at the UA (8), this entity creates a new INVITE request (9) that contains the credentials.

Since the SIP server does not keep a state, and there is also no guarantee that the SIP request arrives at the same SIP server; it is needed to perform the search of the Home SIP server by means of the LIR (10) and LIA (11) messages, as it was done before. The INVITE request is forward to the Home SIP server (12) once its address is known.

The Diameter client in the Home SIP server sends the credentials to the Diameter server in a new Diameter MAR message (13). The Home Diameter server validates the credentials and authorizes the SIP transaction in a Diameter MAA message (14). The SIP server asks for the destination’s address using the LIR (15) and the calling SAAA Home servers answers with a LIA (16) message, which informs about the destination’s address. It might be useful to perform a DNS query to solve this address.

Finally, the Home SIP server forwards the SIP INVITE request to the called Home SIP server (17) as per regular SIP procedures. This entity checks the node that is serving the UA2 using a LIR message (18) and the Home AAA server answers with the address of the server where the UA2 is currently visiting within a LIA message (19). After resolving the address, the INVITE request is forward to the called visited network (20), which contacts with the UA2 (21).

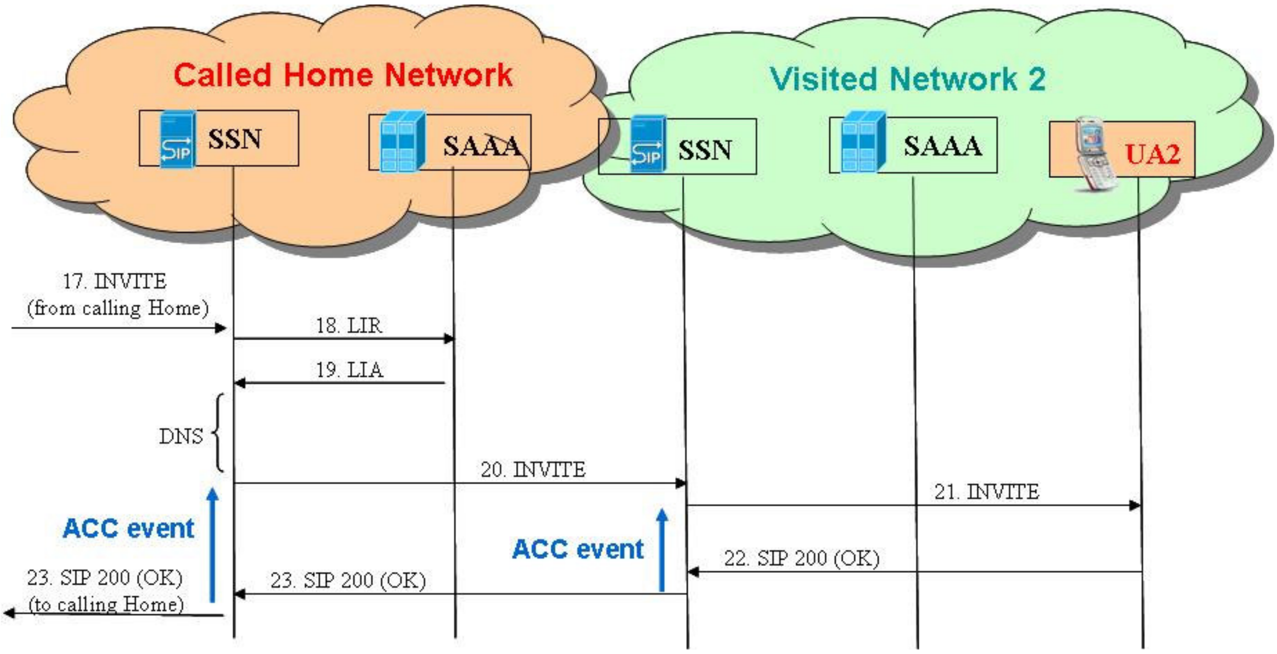


Figure 57: Roaming session setup (2)

Eventually, the session setup is confirmed with a SIP 200 (OK) response (22) that is forwarded to the SIP UA (23) (24) (25) (26). The SIP 200 OK message triggers accounting events. These accounting events inform both the Visited network and the Home network about the session parameters, including type of session, starting time, CODECs in use, caller and callee addresses, etc. The session setup is complete.

10.4 Accounting

The major issues in roaming and mobility lie in the definition of a business model, the setting-up of service level agreements, and billing processes. As these issues are most often not tackled by standardisation work, they could represent a critical bottleneck and significantly delay the commercial launch of roaming solutions.

Following the IMS architecture, the application signaling goes through the calling party's Home network. The Home network is aware of the QoS used and also of the exact time when communication begins and ends. Therefore, its SAAA system could collect accounting information for the whole end-to-end EuQoS session. The Home network will gather both local and global information, acting as a call stateful proxy.

Each network along the path will keep a record of the resources used by the communication at their local SAAA servers (local information). Hence, the same steps as in a non-roaming scenario (see 9.4.1) can be followed from step 3 onwards. The detailed process can be seen in subsection 9.4.2.

11 Use case 3: Non-EuQoS Roaming

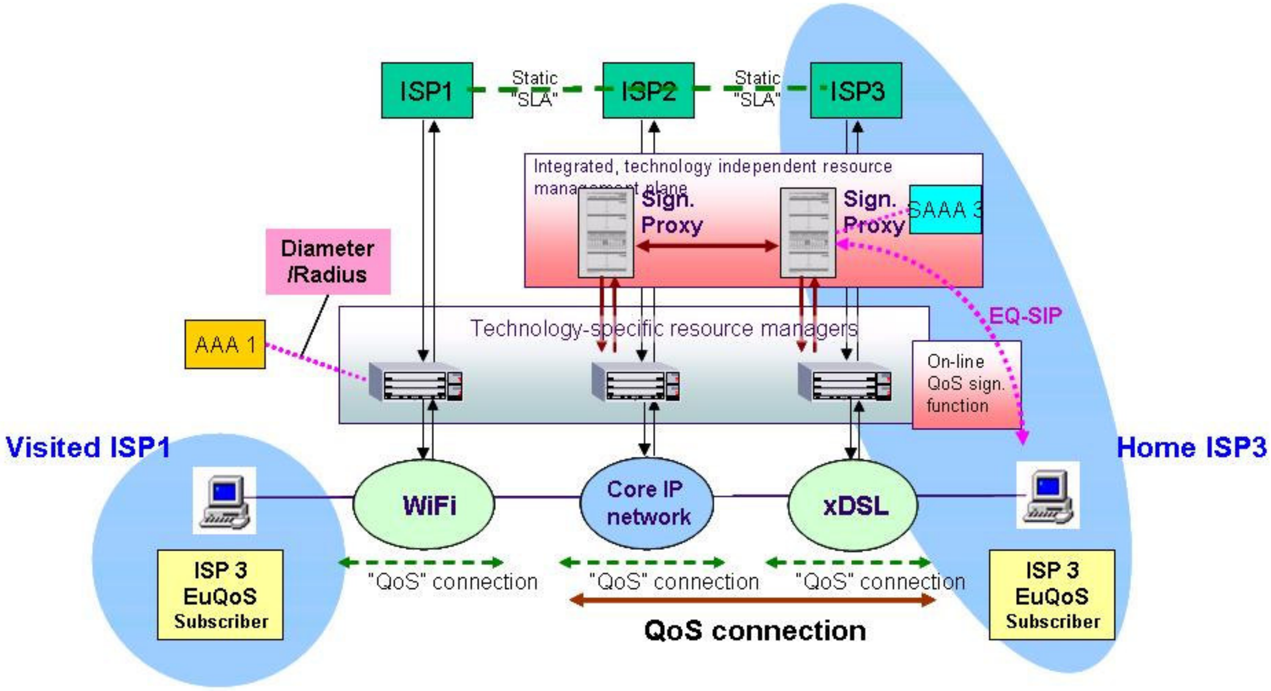


Figure 58: Non - EuQoS diagram

A non-EuQoS aware network is represented in the figure above. The EuQoS user is currently visiting this non-EuQoS Network. Since the EuQoS protocols are unknown to this network, the resource reservation cannot be performed on the whole path. This means that at least at within segment, QoS cannot be guaranteed, which degrades the global system performance to “best effort”. Therefore, only non-EuQoS session will be established in this scenario so this is out of the scope of the project.

12 Conclusions and Future Work

This last chapter gathers conclusions of this thesis. It also presents some ideas for what could be done to complete and improve what has been described in this report.

12.1 Conclusions

This thesis proposes a solution to deal with AAA and charging for the EuQoS system under a roaming environment. DIAMETER has been chosen as the AAA protocol since it offers better performance than RADIUS in roaming networks.

Furthermore, this thesis analyzes several charging models and explains why both online and offline charging mechanisms should be implemented. EuQoS does not define a fixed charging model. Several options are examined in this thesis.

The Session Initiation Protocol (SIP) is used to perform resource reservation. As SIP itself cannot handle a QoS request, therefore SIP was enhanced to become EQ-SIP, which introduces new header fields to manage such reservations. The protocol stack is the same than the one used in SIP. The EQ-SIP protocol is transparent to any non-EuQoS SIP Proxy.

As an existing provider will already be operating its own AAA server, they will be reluctant to use a new AAA structure. Thus, the EuQoS solution should be integrated with the existing AAA architecture in a cost efficient way, so that each service provider can use their existing AAA servers. All inter-domain communication is realized utilizing standard SIP messages, i.e., no separate inter-domain AAA communication is required. This helps to simplify the provider’s infrastructure. However, this increases the complexity of the SIP signalling itself.

One of the main drawbacks of the EuQoS architecture is that it does not provide any QoS reserve in the core network. The initial assumption of an over dimensioned core network works reasonable well when using the GEANT network, but it is likely to be unrealistic when using a lower capacity core network. Furthermore, even measurements using the GEANT network show a small increase in the delay due to the transit thought the GEANT network (around 2ms), which may be significant in some applications.

Authentication and Authorization are performed at the Home network. This introduces extra delay when roaming. Application signalling always needs to pass through the home provider, even if it would have been possible to go directly to the called party without communication with the home provider’s network first. This gives the home provider a complete view of user’s actions. Thus, Authentication and Authorization of any service can be performed at the home network making it possible to control the user’s access. A larger round trip delay is the main drawback associated with this constrain. However, this delay only affects the signalling and not the subsequent media session, so it is unlikely to have any practical or significant impact.

Intermediate networks need not store global accounting data; they should simply collect accounting information about local resource reservations for each EuQoS session. The Local SAAA server for the calling party should also keep an accounting record for the global session parameters. In this way, every network provider involved in the EuQoS session has sufficient information that can be used to charge the Home network for this resource usage.

In summary, as section 10.2.4 demonstrates, the basic architecture proposed in this document scales well in large networks.

12.2 Future work

Security in the AAA access is crucial, thus I would strongly recommend the use of IPSEC, or another secure tunnel that protects any communication related to the AAA server.

The following bullets introduce some areas of interest for continuing the work of this thesis:

- Testing the EQ-SIP protocol via different use cases: The implementation of the EQ-SIP protocol has not been tested yet, thus no results are available. Therefore, creating a usage scenario is necessary to evaluate the EQ-SIP protocol and to determine how the protocol behaves under different workloads of the network and end-points. For instance, a quantitative measurement of the average reservation time, how the protocol modifies the packet delay compared to SIP, and how the roaming solution contributes to the latency of the system. Furthermore, the test could show the optimal configuration of the resources and the network components, and the necessary dimensioning of these entities, the optimal lease time of a service, and, if limited, the maximum number of users that can access the same service simultaneously.

- Analyze how the EQ-SIP protocol can distinguish between different services: Once resources are granted, the EuQoS operator should know the use that the UA makes with them.

- Work towards the standardization of the EQ-SIP protocol.

- Study the system’s vulnerability to a Man in the Middle attack: a malicious entity could place itself inside the signalling path and modify data. Accounting records are especially vulnerable since this malicious entity could change the accounting information.

- It could be interesting to reduce the signalling traffic during the registration phase. One of the main conclusions of [72] is that registration process often takes places more frequently than session setup. With this in mind, it is a good idea to reduce the overhead of the register process as much as possible and delegate to the session setup process any non-time critical action.

- Consider direct refund charging, and how this would be tackled in EuQoS.
- Integrate the EuQoS architecture with an operator's architecture.
- Build and test the proposed schema in a controlled environment.
- Evaluate the system functionality in a real network working under a real load.

13 REFERENCES

[1] Bruno Ferriera, “EuQoS” - End-to-end Quality of Service support over heterogeneous networks, Flor de Utopia - Produções Culturais, Lda., Coimbra, Portugal, 2005, http://www.euqos.org, last accessed Jan 28, 2006.

[2] Cisco Systems, Inc., "Quality of Services", Internetworking Technology Handbook, June 14, 1999, http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/qos.htm, posted February 20, 2003, last modified February 21, 2003.

[3] Jon Postel. “Internet Protocol”, RFC 791, September 1981 http://www.faqs.org/rfcs/rfc791.html

[4] Bernard Aboba, David Mitton, and John Loughney. “AAA: Authentication, Authorization and Accounting”, September 2004. http://www.ietf.org/html.charters/aaa-charter.html

[5] “GÉANT” Gigabit European Academic Network. September 2004. http://www.geant.net/ http://en.wikipedia.org/wiki/GEANT

[6] J. Rosenberg, H. Schulzrinne, G.Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. “SIP: Session Initiation Protocol”, RFC 3261, June 2002. http://www.faqs.org/rfcs/rfc3261.html

[7] M. Handley and V. Jacobson. “SDP: Session Description Protocol”, RFC 2327, April 1998. http://www.faqs.org/rfcs/rfc2327.html

[8] L. Veltri and S. Salsano. “QoS Support for SIP Based Applications in DiffServ Networks”, <draft-veltri-sip-qsip-00.txt> Oct. 2001. It was submitted as internet draft, but never pushed nor presented in IETF.

[9] S. Salsano and L. Veltri. “QoS Control by means of COPS to support SIP based applications”, IEEE Networks. March/April 2002.

[10] L. Veltri, S. Salsano, and D. Papalilo. "SIP Extension for QoS support in Diffserv Networks", Oct 2002 (this was not submitted to IETF, but it was made available at www.coritel.it/projects/qsip).

[11] L. Veltri, S. Salsano, and D. Papalilo. “QoS Support for SIP Based Applications in a Diffserv Network”, IEEE Softcom 2003, October 7-10, 2003, Split, Dubrovnik (Croatia), Ancona, Venice (Italy).

[12] L. Veltri, S. Salsano, and D. Papalilo. "SIP Extensions for QoS support” Internet Draft. October 2002. Expiration: April 2003. File: <draft-veltri-sip-qsip-01.txt>

[13] F. Andreasen. “Session Description Protocol (SDP) Simple Capability Declaration”, RFC 3407. October 2002. Category: Standards Track.

[14] Miikka Pikselkä, Georg Mayer, Hisham Khartabil, and Aki Niemi. “The IMS IP Multimedia Concepts and Services in the Mobile Domain”, John Wiley & Sons, Ltd. 2004, England. ISBN 0-470-87113-X

[15] Christoph Rensing, Hasan, Martin Karsten, and Burkhard Stiller. “A Survey on AAA Mechanisms, Protocols, and Architectures and a Policy-based Approach beyond: Ax” May 2001. Version 1. TIK-Report Nr. 111

[16] Tseno Tsenov, Hannes Tschofenig, Xiaoming Fu, and Eckhart Körner. “Advanced Authentication and Authorization for Quality of Service Signaling” Athens, Greece. September 2005.

[17] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. “RSVP” Resource Reservation Protocol. RFC 2205. September1997. http://www.ietf.org/rfc/rfc2205.txt

[18] Antonio Cuevas, José Ignacio Moreno, Rui Aguiar, Victor Marques, Carlos García, and Ignacio Soto. “Mechanisms for AAA and QoS Interaction”

[19] J. Rosenberg, and H. Schulzrinne. "An Offer/Answer Model with the Session Description Protocol (SDP)", RFC 3264. June 2002. http://www.ietf.org/rfc/rfc3264.txt

[20] G. Camarillo, W. Marshall, and J. Rosenberg. "Integration of Resource Management and Session Initiation Protocol (SIP)", RFC 3312. October 2002. http://www.ietf.org/rfc/rfc3312.txt

[21] G. Camarillo and P. Kyzivat. “Update to the Session Initiation Protocol (SIP) Preconditions Framework” RFC 4032. March 2005.

[22] C. Rigney, S. Willens, A. Rubens, and W. Simpson. “Remote Authentication Dial In User Service (RADIUS)”, RFC 2865. June 2000. http://www.ietf.org/rfc/rfc2865.txt

[23] W. Simpson. “The Point-to-Point Protocol (PPP)”, RFC 1661. July 1994. http://www.ietf.org/rfc/rfc1661.txt

[24] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, and J. Arkko. “Diameter Base Protocol”, RFC 3588. September 2003. http://www.ietf.org/rfc/rfc3588.txt

[25] Dorgham Sisalem Jiri Kutham. “Inter-domain Authentication and Authorization Mechanisms for Roaming SIP Users”, January 2005. Berlin, Germany.

[26] J. Vollbrecht, P. Calhoun, S. Farrell, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdr ege, and D. Spence. “AAA Authorization Framework”, RFC 2904. August 2000. http://www.ietf.org/rfc/rfc2904.txt

[27] C. de Laat, G. Gross, L. Gommans, J. Vollbrecht, and D. Spence. “Generic AAA Architecture”, RFC 2903. August 2000. http://www.ietf.org/rfc/rfc2903.txt

[28] C. Perkins. " IP Mobility Support for IPv4", RFC 3344. August 2002. http://www.ietf.org/rfc/rfc3344.txt

[29] C. Perkins. "IP Encapsulation within IP", RFC 2003. October 1996. http://www.ietf.org/rfc/rfc2003.txt

[30] C. Perkins. "Minimal Encapsulation within IP", RFC 2004. October 1996. http://www.ietf.org/rfc/rfc2004.txt

[31] J. Solomon, and S. Glass. "Mobile-IPv4 Configuration Option for PPP IPCP", RFC 2290. February 1998. http://www.ietf.org/rfc/rfc2290.txt

[32] S. Glass, T. Hiller, S. Jacobs, and C. Perkins. ”Mobile IP Authentication, Authorization, and Accounting Requirements”, RFC 2977. October 2000. http://www.ietf.org/rfc/rfc2977.txt

[33] S. farell, J. Vollbrecht, P. Calhoun, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege, and D. Spence. “AAA Authorization Requirements”, RFC 2906. August 2000. http://www.ietf.org/rfc/rfc2906.txt

[34] F. Johansson and T. Johansson. “Mobile IPv4 Extension for Carrying Network Access Identifiers”, RFC 3846. June 2004. http://www.ietf.org/rfc/rfc3846.txt

[35] Funk Software, Inc. white paper. “The Role of RADIUS/AAA in 3G/Mobile IP-Based Environments”, 2005 Cambridge, England.

[36] **http://en.wikipedia.org/wiki/Authentication**

[37] B. Aboba, J. Arkko, and D. Harrington. “Introduction to Accounting Management”, RFC 2975. October 2000. http://www.ietf.org/rfc/rfc2975.txt

[38] Håkan Ventura. “Diameter next generation’s AAA protocol”, April 2002. Linköping, Sweden.

[39] N. Brownlee and A. Blount. “Accounting Attributes and Record Formats”, RFC 2924. September 2000. http://www.ietf.org/rfc/rfc2924.txt

[40] "Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON); Inter-domain pricing, authorization, and usage exchange", TS 101 321 V1.4.2, December 1998.

[41] C. Rigney. "RADIUS Accounting", RFC 2866. June 2000. http://www.ietf.org/rfc/rfc2866.txt

[42] ITU-T Q.825: *Specification of TMN Applications at the Q3 Interface: Call Detail Recording;* Recommendation Q.825, Geneva Switzerland, 1998.

[43] S. A. Cotton (edt.): *Network Data Management – Usage (NDM-U) for IP-Based Services;* IPDR Specification Version 1.1, June 2000.

[44] Dan Spears and Douglas N. Zuckerman. “Call Detail Records for UNI 1.0 Billing”, *April 200.* Fremont, CA, USA. *http://*www.oiforum.com

[45] R. Calhoun, G. Zorn, P. Pan, H. Akhtar, "Diameter Framework Document", <draft-ietf-aaa-DIAMETER-framework-01.txt>, IETF work in progress, March 2001.

[46] D. Mitton, M. St.Johns, S. Barkley, D. Nelson, B. Patil, M. Stevens, and B. Wolff. “Authentication, Authorization, and Accounting: Protocol Evaluation”, RFC 3127. June 2001.

[47] P. Calhoun, G. Zorn, D. Spence, and D. Mitton. “Diameter Network Access Server Application”, RFC 4005. August 2005. http://www.faqs.org/rfcs/rfc4005.html

[48] T. Hiller and G. Zorn. “Diameter Extensible Authentication Protocol (EAP) Application”, <draft-ietf-aaa-eap-01.txt> March 2003.

[49] Perkins, Calhoun, "AAA Registration Keys for Mobile IP", draft-ietfmobileip-aaa-key-01.txt, IETF work in progress, January 2000

[50] J. Arkko, P. Calhoun Glen Zorn "Diameter Accounting Extensions”, <draft-ietf-aaa-DIAMETER-accounting-01.txt> Internet-Draft. March 2001. http://www3.ietf.org/proceedings/01mar/I-D/aaa-DIAMETER-accounting-01.txt

[51] S. Kent and R. Atkinson. "Security Architecture for the Internet Protocol", RFC 2401. November 1998.

[52] S. Blake-Wilson, M. Nystrom, D. Hopwood, J. Mikkelsen, and T. Wright, "Transport Layer Security (TLS) Extensions", RFC 3546. June 2003.

[53] P. Calhoun, S. Farrell, and W. Bulley. "Diameter CMS Security Application”, <draft-ietf-aaa-DIAMETER-cms-sec-04.txt> March 2002.

[54] G. Q. Maguire Jr., “Lecture notes of the course 2G1325/2G5564 Practical Voice Over IP (VoIP): SIP and related protocols”, Module 14 (AAA and QoS for SIP): 344 of 365, April 2005. https://www.it-univ.se/courses/2G1325/VoIP-2005.pdf

[55] Gonzalo Camarillo. “Session Initiation Protocol (SIP) Overview”, http://internetng.dit.upm.es/ponencias-jing/2005/SIP.pdf last acceded in October 2005.

[56] T. Berners-Lee, R. Fielding, and L. Masinter, “Uniform Resource Identifier (URI): Generic Syntax”, IETF RFC 3986, January 2005.

[57] G. Q. Maguire Jr., “Lecture notes of the course 2G1325/2G5564 Practical Voice Over IP (VoIP): SIP and related protocols”, Module 3: 131 of 194, March 2004.

[58] L. Dang, C. Jennings, and D. Kelly. Practical VoIP using VOCAL. O'Reilly & Associates Inc. ISBN 0–596–00078–2. July 2002.

[59] A. B. Roach, B. Campbell, and J. Rosenberg. "A Session Initiation Protocol (SIP) Event Notification Extension for Resource Lists", <draft-ietf-simple-event-list-07>. December 15, 2004. Expires: June 15, 2005. http://www.ietf.org/internet-drafts/draft-ietf-simple-event-list-07.txt

[60] A. Gulbrandsen, P. Vixie, and L. Esibov, "A DNS RR for Specifying the Location of Services (DNS SRV)", IETF RFC 2782, February 2000.

[61] H. Schulzrinne and B. Volz. "Dynamic Host Configuration Protocol (DHCPv6) Options for Session Initiation Protocol (SIP) Servers", RFC 3319. July 2003. http://www.faqs.org/rfcs/rfc3319.html

[62] Juniper Networks. "Evolution of Session Border Controllers" white paper. CA, USA. April 2005 http://www.juniper.net/solutions/literature/white_papers/200119.pdf

[63] 3GPP TR 21.905: Vocabulary for 3GPP Specifications.

[64] M. Baugher, D. McGrew, M. Naslund, E. Carrara, and K. Norrman. "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004. http://www.ietf.org/rfc/rfc3711.txt

[65] J. Arkko, E. Carrara, F. Lindholm, M. Naslund, and K. Norrman. "MIKEY: Multimedia Internet KEYing", RFC 3830. August 2004. http://www.ietf.org/rfc/rfc3830.txt

[66] H. Hakala, L. Mattila, J-P. Koskinen, M. Stura, and J. Loughney. "Diameter Credit-Control Application", RFC 4006. August 2005. http://www.ietf.org/rfc/rfc4006.txt

[67] 3GPP TS 32.225, Telecommunication management; Charging management; Charging data description for the IP Multimedia Subsystem (IMS), http://www.3gpp.org/ftp/Specs/html-info/32225.htm

[68] GSM Association Official Document IR.61. "WLAN Roaming Guidelines" (also known as Inter-Operator Handbook) http://www.gsmworld.com/documents/wlan/ir61.pdf August 2004.

[69] Sami Keski-Kasari, Karri Huhtanen, and Jarmo Harju. "Applying Radius-based Public Access Roaming in the Finnish University Network (FUNET)" Tampere University of Technology, Finland.

[70] Myers, Ankney, Malpani, Galperin, Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol (OCSP)", RFC 2560, June 1999.

[71] P. Calhoun, W. Bulley, S. Farrell, "DIAMETER Strong Security Extension", draft-calhoun-DIAMETER-strong-crypto-03.txt, IETF work in progress, April 2000.

[72] Raúl García Hijes. “Corporate Wireless IP Telephony” Section 8.5.2. MIP overhead, July 2005. Stockholm, Sweden.

14 Appendix

A. Press Release:

“ A new EU Integrated Project EuQoS established to bring end to end Quality of Service support for applications. The EuQoS consortium announces its intention to help resolve the outstanding Quality of Service (QoS) issues in the telecommunications industry.

It is necessary to resolve these issues in order that real-time applications can be supported by the Internet. Such applications include voice, video-conferencing, video-streaming, educational, tele-engineering and medical applications.

A consortium of 24 partners from industry and academia - lead by Telefonica I&D - have come together to create the EuQoS consortium as part of the EU Framework Programme 6. The goal of the EuQoS project is to build a Flexible and Secure QoS Assurance System (the EuQoS System) and to deploy it in a pan-European test environment.

José Enríquez Gabeiras, CEO of the consortium said “The idea is to build a European wide research test-bed to deploy QoS solutions. This test bed will be used to support end-user applications over multiple access network infrastructure test-beds, interconnected by the GÉANT backbone network. The test-bed will include research networks from ten different countries.”

Martin Potts CTO of the consortium said that “EuQoS will research, develop and integrate state-of-the-art QoS technologies in order to prepare for future QoS upgrades in European telecommunications infrastructure and deliver roadmaps and strategic guidance to industry.”

Donal Morris, Chairman of the consortium said that “It is important to invest in technology for QoS in order that new applications such as video conferencing can be supported, with adequate quality for customers using both broadband wireline and wireless networks.”

The total budget of the project is over 16mE with 9.5mE being provided by the European Commission. The project will last 36 months and will provide a first major demonstration of its QoS system within 18 months.

The EuQoS project will be based on large scale experimentation on a pan-European network in order to promote interoperability between heterogeneous technological network domains. This will enable the interoperability of QoS solutions across xDSL, UMTS, WiFi, Satellite, Ethernet, and DWDM/MPLS access networks giving rise to new application usage and a greater volume of premium traffic on the Internet.

About EuQoS

EuQoS is an Integrated Project under the EU's Framework Programme 6. The goal of the EuQoS project is to build a Flexible and Secure QoS Assurance System (the EuQoS System) and to deploy it in a pan-European test environment. The consortium contains 24 partners from industry and academia including the following network operators: France Telecom R&D, ERA, Portugal Telecom Inovação, Telefonica I&D, and Polish Telecom.

From industry the following solution and consulting providers are supporting the project: Datamat, Siemens SBS, Soluziona, Ericsson, Juniper, Martel, PointerCom, Red Zinc, Silogic, and Telscom

A number of key research organisations are providing an academic input: LAAS-CNRS, NICTA, University of Bern, Technical University of Catalonia/UPC, University of Coimbra, University of Paderborn, University of Pisa/ CPR, University of Rome/CRMPA, WUT-Wars aw University of Technology and HDES.

EuQoS website www.euqos.org “

EU Integrated Project EuQoS - Madrid, 1 September 2004

B. Definitions

- **Home Network:** This is the network of an EuQoS service provider with whom the user maintains an account relationship.
- **Local Network:** This is the network of an EuQoS service provider with whom the user initiates an EuQoS connection. Where roaming is implemented, the local network may be different from the home network.
- **Visited Network:** This is the network of an EuQoS service provider with whom the user initiates an EuQoS connection when roaming.
- **Roaming:** An EuQoS user is roaming whenever he or she is accessing the EuQoS system from a different network (Visited/Local Network) from where he/she originally made the contract with his/her EuQoS provider (Home Network).
- **SIP Proxy:** A proxy server receives a request and then forwards it towards the current location of the callee -either directly to the callee or to another server that might be better informed about the actual location of the callee.
- **SIP Redirect:** A redirect server receives a request and informs the caller about the next hop server. The caller then contacts the next hop server directly.
- **SIP User Agent:** A logical entity in the terminal equipment that is responsible for generating and terminating SIP requests. Consist of two parts: User agent client and user agent server:
    - **User agent client (UAC):** The client part of SIP user agent. All SIP requests are initiated by this part of SIP UA.
    - **User agent server (UAS):** This is another part of the UA. When a request is received by the UA, the UAS is this part which processes the request and sends a response back to the client, who initiated the request.
- **SIP Registrar:** To assist SIP entities in locating the requested communication partners SIP supports a further server type called register server. The register server is mainly thought to be a database containing locations as well as user preferences as indicated by the user agents.
- **Online charging:** It is a charging process where entities, such as an application server (AS), interact with the online charging system. The online charging system in turn interacts in real

time with the user's account and controls or monitors the charges related to service usage.

- **Offline charging:** It is a charging process where charging information is mainly collected after the session and the charging system does not affect in real time the service being used. In this model a user typically receives a bill on a monthly basis, which shows the chargeable items during a particular period.

- **Subscription:** A subscription describes the commercial relationship between the subscriber and the service provider.

- **Billing:** A role played by a system whereby detail records received from Charging Control are transformed into bills requiring payment.

- **Charging Control:** The role that processes charging input resulting in charging information. Processing includes mediation, rating, spending and credit control. The processing by Charging Control may result in charges raised against subscriber account(s).

- **Credit Control:** The process of checking the account that credit is available, reservation of credit, reduction of credit from the subscriber account when service is completed and refunding of reserved credit not used.

- **Credit reservation:** The term reservation denotes a temporary deduction of an account that may be later on permanent. The purpose is to secure sufficient funding of service usage.

- **CDR:** Charging Data Record, or Call Detail Record. A formatted collection of information about a chargeable activity (time initiated, duration, volume, and many other details) for use in accounting, billing, charging, settlement and statistics. For each party to be charged, for parts of or all charges of a chargeable activity a separate CDR is generated. For example more than one CDR may be generated for a single chargeable activity, e.g. because of its long duration, or because more than one charged party is to be charged.

- **Event Charging:** Charging of a service that comprises a single delivery unit from charging perspective, e.g. donating money to a welfare organization or paying for sending a SMS.

- **Rating:** Rating is a process of computing a price. This process considers charging input (volume, duration, category, etc.), configuration data (tariff tables etc.) and a context that is obtained from other sources (time of day, location of the user etc.).

- **Service Provider:** A service provider is either a network operator or another entity that provides services to a subscriber (e.g. a MVNO). Note: This is a 3GPP definition [63].
- **Serving Element:** The role played by a system interacting with charging control for the purpose of charging. Examples for serving elements are GGSN, MSC and MMS-C.
- **Session charging:** Method performing subsequent charging requests during delivery of a service. Charging of a voice call is a typical example for usage of session charging.
- **Spending Control:** A service providing the user with spending information and safeguarding him against uncontrolled spending.
- **Subscriber:** Subscriber is the individual or organization responsible to settle an account. Note that the subscriber that owns the account may or may not be the same person as the user.
- **Tariff:** Price per unit - a unit could be time, volume, event etc.
- **User:** The individual or a system that consumes a service.
- **Diameter:** Diameter is an evolution of RADIUS where the connection to a specific service for dial in services is removed. The Diameter base protocol is intended to provide an AAA framework for applications such as network access or IP mobility. Diameter is also intended to work in both local AAA and roaming situations. On top of DIAMETER you have to add applications for specific services.
- **Prepaid:** A payment option where the customer pays in advance for services that they may use.
- **Postpaid:** A payment option where the user pays after the service is delivered.

C. Abbreviations

3GPP	Third Generation Partnership Project
AAA	Authentication, authorization and accounting
ADSL	Asynchronous Digital Subscriber Line
AS	Autonomous system
ASIG	Application signaling
ATM	Asynchronous transfer mode
AVP	Attribute value pair
A-SSN	Application signaling and service negotiation
CDR	Charging Data Record/Call Detail Record
CMS	Cryptographic Message Syntax
COPS	Common Open Policy Service
DFD	Data flow diagram
DHCP	Dynamic Host Configuration Protocol
DNS	Domain name system
DSL	Digital Subscriber Line
ETSI	European Telecommunications Standards Institute
FQDN	Fully qualified domain name
GEANT	Gigabit European Academic Network
GGSN	Gateway GPRS Support Node
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
HTTP	Hyper Text Transfer Protocol
IETF	Internet Engineering Task Force
IMS	IP Multimedia Subsystem
IP	Internet Protocol
IPsec	Internet Protocol security
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
ISDN	Integrated Services Digital Network
ISP	Internet Service Provider
ITU-T	International Telecommunication Union – Telecommunication Standardization Sector
LIA	Location-Info-Answer
LIR	Location-Info-Request
MAA	Multimedia-Multimedia-Answer
MAR	Multimedia-Auth-Request
MIP	Mobile IP
NASREQ	Network Access Server Requirements
QoS	Quality of service
RADIUS	Remote Authentication Dial In User Service
RM	Resource manager
ROAMOPS	Roaming operations
RSVP	Resource Reservation Setup Protocol
SA	Security association

SAAA	Secure authentication, authorization and accounting server
SCTP	Stream Control Transmission Protocol
SDP	Session Description Protocol
SHA	Secure Hash Algorithm
SIP	Session Initiation Protocol
SLA	Service-level agreement
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UA	User Agent
UAA	User-Authorization-Answer
UAC	User Agent Client
UAR	User-Authorization-Request
UDP	User Datagram Protocol
UE	User equipment
UMTS	Universal Mobile Telecommunications System
URI	Uniform resource identifier
URL	Universal resource locator
VoIP	Voice over IP
XML	Extensible Markup Language

COS/CCS 2006-7

www.kth.se