Quality of Service and security negotiation for autonomous management of Next Generation Networks

Mohamed Aymen Chalouf
LaBRI Laboratory, University of Bordeaux
351 cours de la Libération, F-33405, Talence (France)
Tel: 33-540-003-501 E-mail: chalouf@labri.fr

Nader Mbarek
LE2I Laboratory, University of Bourgogne
Aile des sciences de l'ingénieur, 21078, Dijon (France)
Tel: 33-380-395-910 E-mail: Nader.Mbarek@u-bourgogne.fr

Francine Krief
LaBRI Laboratory, University of Bordeaux
351 cours de la Libération, F-33405, Talence (France)
Tel: 33-540-003-501 E-mail: krief@labri.fr

Received: June 21, 2011 Accepted: August 11, 2011 DOI: 10.5296/ npa.v3i2.730

Abstract

Based on the IP technology, the next generation networks (NGN) must overcome the main drawbacks of this technology consisting in the lack of quality of service (QoS), security and mobility management. To ensure a service offer in an NGN environment, a protocol for negotiating a service level can be used. However, most of the existing negotiation protocols allow the establishment of a service level which includes only QoS. As for security and mobility, they were often not covered by these negotiations, and therefore managed independently. However, securing a service can cause degradation of the QoS, and the mobility of a user can change the service needs in terms of QoS and security. Thus, we need to simultaneously manage QoS and security while taking into account user's mobility. In this context, we propose an end-to-end NGN architecture with several autonomous domains, where each domain is under the authority of an autonomic domain manager. We provide those autonomic managers with a negotiation capability to achieve an agreement between the different domains involved in the transport of various offered services. This agreement may cover different aspects such as QoS, security and takes into account the user profile. The negotiation process will be based on SLNP (Service Level Negotiation Protocol) which uses Web Services technologies in order to enable interoperability between the managers of the different autonomous domains involved in the service transport.

Keywords: Next generation networks, Autonomous management, Service level negotiation, Quality of Service, Security, User profile.

1. Introduction

NGN networks enable different access networks using heterogeneous technologies to provide a global connectivity and various services offer. In such environment, we notice a growing complexity in infrastructure management where traditional techniques with human intervention are no longer efficient. Indeed, the more the number of inter-connected systems is important, the more it is difficult to anticipate the interactions between their components. One of the stakes of the coming years is the deployment of a new paradigm, which will make it possible to bring autonomy in the Information Technology (IT) infrastructures thanks to a new management concept called Self-Management.

In the NGN with self-management environment, many interactions could take place when delivering or consuming a service. With the emergence of IP networks and the increasing number of applications requiring a high level of QoS, security and mobility the guarantee of an end-to-end service level become increasingly critical. Providing autonomous systems with the capability of service level negotiation, which takes into account the QoS as well as the security and the mobility, is a very challenging task.

In this context, we had specified a negotiation protocol which allows the dynamic negotiation of a service level including simultaneously QoS and security. This negotiation protocol is based on the use of the Web Services (WS) technologies in order to provide the different negotiation parts with interoperability. Thus, the negotiation initiation can be based on the user profile, which will optimize and automate the negotiation process.

The remainder of this paper is organized as follow: section 2 describes the state of the art in terms of networks architecture evolution, autonomous management, service level negotiation and user profile. In section 3, we present an autonomous management architecture. Section 4 describes a framework for QoS and security negotiation for mobile users. Then, section 5 presents a test bed for our service level negotiation proposition with an implementation of SLNP. Lastly, section 6 concludes the paper and points out future work.

2. State of the art

In this section, we introduce the next generation networks architecture and the autonomous management concepts which constitute the general context of the presented work. Then we define the service level negotiation and how this mechanism can meet the needs of NGNs management by enabling the different actors to provide some guarantees (QoS, security, etc.). Finally, we present some results relating to user profiles that help us in the definition of the user profile on which the negotiation process will be based, in order to provide the required guarantees in an NGN environment.

2.1 Network architecture: evolution and needs

2.1.1 Network convergence

Until the 90s, there were three separate and independent networks: Broadcast, Telecommunication and Internet. Each network has its own infrastructure and interconnection between these networks was very limited. Then, in recent years, these networks have much evolved. Indeed, the Telecommunication network becomes digital, wireless and mobile. The Broadcast network is migrating to digital radio and television. As for IP network, it provides new services like multimedia services including the transmission of audio and video flows. Thus, the boundaries between the existing three networks tend to disappear and services are widespread on all networks. For example, Internet and TV are available on Telecommunication networks, phone communications may be transported over the Internet network, etc.

This movement leads to the emergence of the concept of network convergence, which aims to specify a global framework for grouping the existing networks under a single architecture: Next Generation Networks (NGN).

2.1.2 NGN architecture

The NGN represents the next generation of networks which can achieve full convergence between the traditional networks with different types of services into a single architecture. Thus, they allow the coexistence of all services that would be accessed by all users using any access network and any equipment.

The NGN architecture [1], presented in Figure 1, is mainly characterized by a horizontal decomposition into two functional layers; the "Transport" layer which uses IP to provide the various components of an NGN with connectivity, and the "Service" layer that provides basic functionalities (e.g. registration, notification, presence, etc.) for the different services.

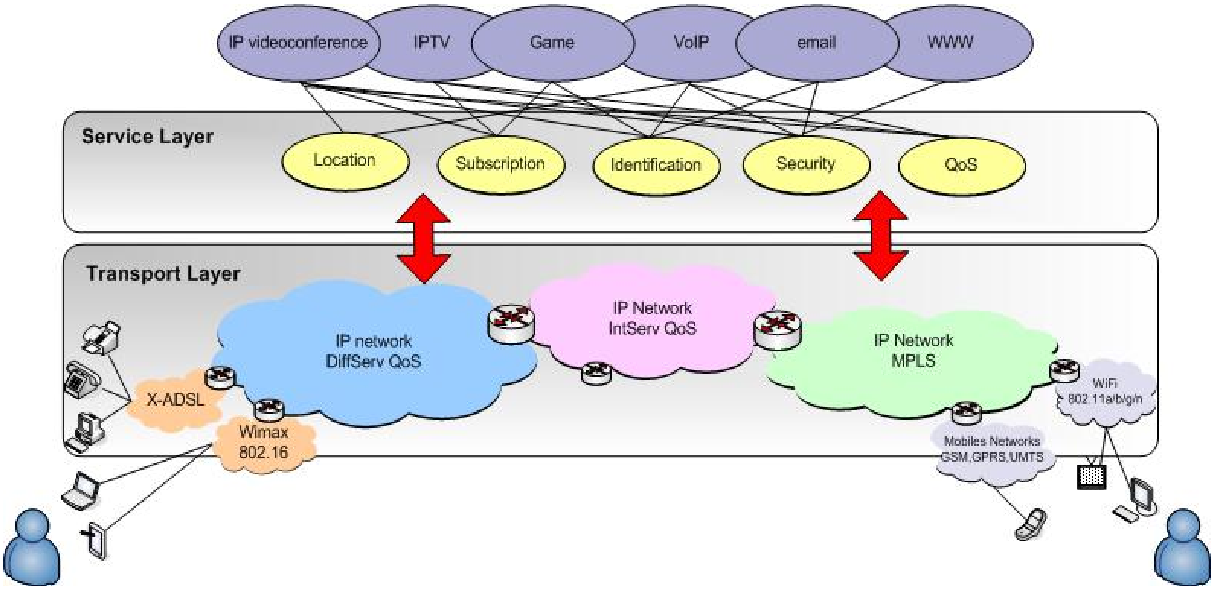


Figure 1. NGN architecture

After this horizontal decomposition, the NGN architecture is also characterized by a differentiation between access networks and core network, the first aim to concentrate traffic of individual users in order to forward it to central entities. While, the second is designed to route large volumes of traffic between several core entities. In Figure 1, we can note the diversity of technologies that can allow a user to access services. We also note that the core network is composed of one or many autonomous domains where each domain includes a set of nodes responding to a single administrative authority. In this work we consider that these domains are autonomous and we explain, in the following section, the concepts and the characteristics of such systems.

2.1.3 Autonomous systems: characteristics and concepts

Within the NGN context, many indicators such as wireless networks quantitative explosion and the emergence of heterogeneous new technologies are making Information Technologies (IT) infrastructures management more and more complex with traditional techniques. The most adequate way to manage this complexity became a paramount subject. To meet this need, a consensus seems to emerge and make various positions converge towards concepts of self-management [2] with technologies that make it possible for a system to manage itself and become autonomic without human interventions.

The vision for bringing autonomy in IT infrastructures is the creation of self-management systems (autonomous systems without human intervention) to cope with increasing complexity and excessive maintenance costs while preparing the road to satisfy the pervasive computing needs of NGN architecture in the future. Such autonomous systems are able to be self-organized. Networks become a collection of interconnected self-governed entities, which do not need the human intervention, except for high-level directives specification. Thus, autonomous system management details of software and hardware components are transparent to the administrators.

The starting point of this new paradigm is biological systems and in particular, the autonomic nervous system to which we partly owe the term Autonomic Computing [3]. Like humans with their autonomic nervous system, networks with their autonomic entities must be reliable and offer guaranties of availability, security, safety, maintainability: that is to say achieve dependability [4]. This new management concept makes it possible thanks to a holistic approach where all research fields contribute in providing networks and systems with a global autonomy.

Although the objectives list of the self-management concept was extended since 2001 (year of this new paradigm birth), the main objectives for autonomous systems are Self-configuring, Self-optimizing, Self-healing and Self-protecting [2]. To achieve those objectives, autonomous systems have a detailed knowledge of their internal state as well as their environment [5] thanks to a continuous monitoring of eventual changes that could affect their components. Detecting changes induce the autonomous system to adjust its resources and the monitoring continues to determine if the new measures satisfy the desired performance. That is the closed control loop of self-management systems. It enables autonomous systems to make adequate decisions while conforming to global objectives without human interventions thanks to measurements collected from its resources. This closed control loop is implemented by autonomic managers, which control managed resources thanks to sensors and effectors manageability interfaces [6].

2.2 Service level negotiation

2.2.1 Service level definition

When a client wants to consume a service, he must negotiate with the service provider a contract called Service Level Agreement (SLA). This contract defines the service that the client should receive [7], by specifying some aspects such as availability, performance, actions in case of failure or dysfunction of service and billing type. An SLA contains both technical and non-technical parameters. The technical parameters constitute the negotiable part of the SLA, and are grouped together in a specification called Service Level Specification (SLS) [8]. Thus, the SLS is a set of parameters and their corresponding values which allow the definition of the service offered to a given traffic. These parameters can cover various aspects such as QoS, security and mobility.

Hence, a service offering in NGNs could be defined through a service level represented by an SLA which is a contract between the service provider and the client. To guarantee an end-to-end service level, the managers of the different domains implied in a service offer must agree on the SLS parameters. This need can be met through the use of a signaling protocol enabling the negotiation of a service level.

2.2.2 Service level negotiation protocols

Several protocols have been proposed to enable dynamic negotiation of service level. These protocols can be classified into two categories [9]. First, the protocols which constitute extensions of other signaling protocols such as SLS IPCP (Internet Protocol Control Protocol) [10] and COPS-SLS (Common Open Policy Service) [11]. Then, the protocols that are specified especially for service level negotiation like QoS-GSLP (QoS Generic Signaling Layer Protocol) [12], SrNP (Service Negotiation Protocol) [13] and DSNP (Dynamic Service Negotiation Protocol) [14]. The SLNP protocol, considered in this paper, belongs to the second category. In fact, it was specified to enable the negotiation of service level in an interoperable manner through the use of Web Services [15]. Therefore, it is an out-of-band signaling protocol which provides domain managers with interoperability.

The majority of the above mentioned protocols are only QoS aware because QoS represented the most immediate service level need with the development of new services such as VoIP, VoD, IPTV, etc. As for security and mobility, they were often excluded from the service level considered is the negotiation. Since it is essential to consider both security and QoS in an NGN autonomous environment service offer, we believe that SLNP is well suited to perform the service level negotiation in such environments thanks to Web Services interoperable technologies usage. Indeed this protocol is able to associate security and QoS in the negotiated service level while taking into account user mobility. Taking into account the mobility of users in a service offer is performed by using a user profile for the negotiation. Thus, in the next section, we describe some related works that helped us to define the user profile on which the SLNP negotiation will be based.

2.3 User profile

A user profile is a set of various data relating to a user. It is a knowledge source which can include identification information and additional information on the communication context, allowing a better adaptation to the whole connection environment, such as user preferences, screen size, resources availability, etc. In this part, we provide some examples of user profile proposed in standards and research works.

2.3.1 Some standardized user profiles

- *Composite Capability/Preferences Profiles (CC/PP):* The CC/PP Profile [16] was defined in order to specify a standard manner for describing terminal capabilities and user preferences. This profile, expected to be used with HTTP, often refers to a transmission context, and enables the content adaptation. The structure of the CC/PP profile is characterized by a two-level hierarchy. The first level contains one or more of the three defined components: “Terminal Hardware”, “Terminal Software” and “Terminal Browser”. At the second level, we find the attributes, included in the mentioned components, which enables describing the terminal capabilities and user preferences.

- *Usage Environment Description (UED):* The MPEG-21 standard [17] was defined in order to specify a standardized architecture for multimedia content delivery. Part 7 of this standard [18] defines a set of tools to perform the adaptation of media stream. Among these tools, the UED offers standardized description of user characteristics and environment. This description covers four aspects: user characteristics (e.g. display, color, content and language preferences), terminal capabilities (e.g. supported codec, screen size, storage capacity), network characteristics (e.g. maximal capacity, error connection, bandwidth, loss rate, jitter, signal strength) and environment characteristics (e.g. location, audio noise degree, brightness degree). In this case, defined user profile is quite general, since it contains four different types of parameters which could be very interesting in a service level negotiation context. Therefore, most of these profile components: user, terminal and network are considered in the user profile for our SLNP negotiation in NGNs.

2.3.2 A user profile for QoS negotiation

There are several research works that deal with QoS management, based on the application needs and user characteristics like those presented in [19], [20] and [21]. For example, a user profile for QoS negotiation has been defined when specifying a “smart” interface that allows users to negotiate QoS [19]. In this work, the identified user profile parameters were classified into three categories: user characteristics, user preferences and application characteristics. These parameters cover user and application and enable the definition of the QoS level needed for a communication. Since our SLNP negotiation concerns both QoS and security, there are other aspects that will be considered like user preferences regarding security, security characteristics of the access network, etc. This leads to define a more general user profile that includes several types of parameters (section 4.3).

3. Autonomous management architecture

3.1 Global architecture

We propose an autonomous management architecture (Figure 2) made of several Autonomous Systems (AS) to guarantee a service level for traffic flows generated from an ingress AS to an egress AS. Every autonomous system uses one High-level Autonomic Manager (HAM) to control several Low-level Autonomic Managers (LAM) when negotiating a service level with the corresponding HAMs of other AS. A LAM is called Border LAM (B-LAM) when it is responsible of an access network (WiFi, WiMAX, Wired, etc.) according to the NGN concepts and a Core LAM (C-LAM) when situated in the core network of NGN architecture (cf. section 2.1.3). Two kinds of interactions could take place between those components to provide our architecture with autonomy while offering an end-to-end QoS and security guarantee. The following sections describe those interactions while defining the architecture components.

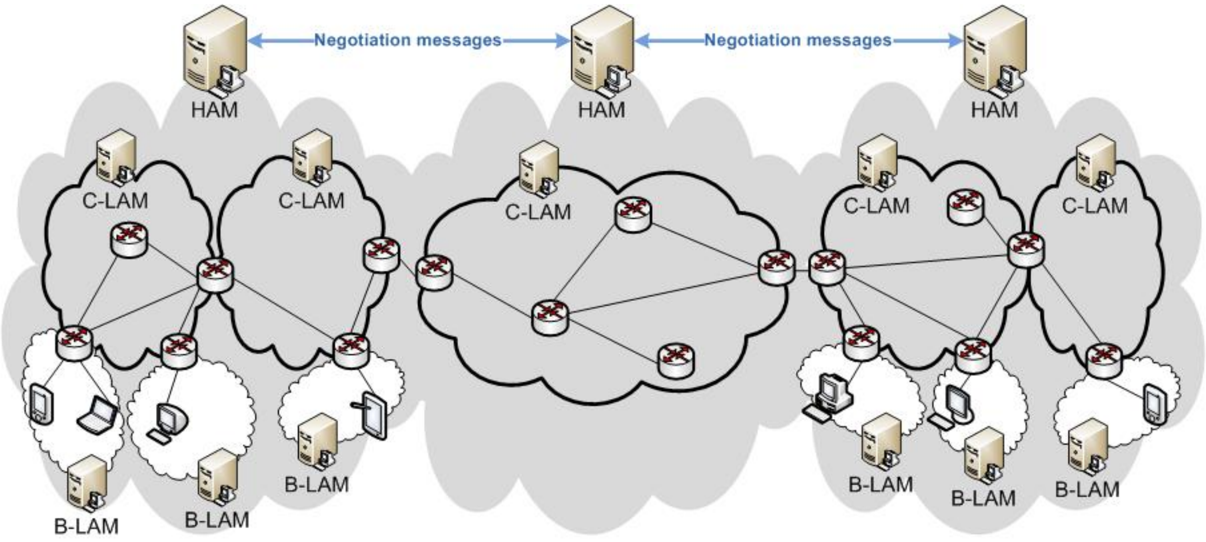


Figure 2. Autonomous management architecture

3.2 High-level Autonomic Manager (HAM) for horizontal interaction

A High-level Autonomic Manager uses a negotiation protocol to communicate with other HAMs to achieve an agreement on a service level. Besides, it controls one or more LAMs thanks to a standardized manageability interface using Web Services technologies.

We consider the HAM as a component of the NGN service layer. Indeed, the HAM is responsible of service level guarantee thanks to a first kind of interaction. It is an inter AS horizontal interaction where an ingress Autonomous System thanks to its High-level Autonomic Manager (HAM), initiates a peer to peer negotiation process with the following HAMs until arriving to the High-level Autonomic Manager of the egress Autonomous System. The communication between HAMs is based on the SLNP negotiation protocol that uses Web services technologies. All these HAMs must previously publish their Web services in the

UDDI registry to make the negotiation process possible. Once the negotiation procedure finished successfully, every AS and precisely every HAM is responsible of the service level guarantee within the corresponding AS. This guarantee will be possible thanks to a second kind of interaction. We specify in section 4 the SLNP protocol that enables the negotiation process to take place in the horizontal interaction.

3.3 Low-level Autonomic Manager (LAM) for vertical interactions

A Low-level Autonomic Manager controls one or more managed resources thanks to the same kind of interface that a HAM uses for managing it (i.e Sensor and Effector manageability interfaces). Managed Resources (MR) could be hardware or software entities (server, router, database, etc.). One or more managed resources expose their manageability capabilities thanks to one manageability interface made of Sensor and Effector interfaces. Sensor interface enables the HAM and the LAM to request or receive a monitoring information notification from, respectively, the LAMs and the MRs which are under their control. Effector interface gives the HAM and the LAM the means to perform actions for modifying the behavior of the entities under their control.

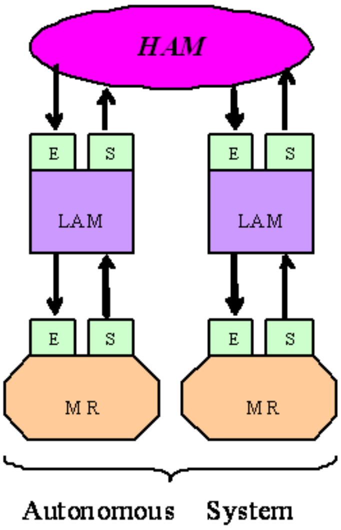


Figure 3. Vertical interactions

LAMs are responsible of service level guarantee within their respective local autonomous domains thanks to a second kind of interactions (Figure 3). It is an intra AS vertical interactions where High-level Autonomic Manager provides the Low-level Autonomic Managers with the negotiated service level so that they modify the configuration of their managed resources according to the service level parameters values. The manageability interface (Effector and Sensor) that enables these vertical interactions between a HAM and a LAM but also a LAM and the corresponding managed resources must be built in conformance with open standards. In that way, standardization efforts are made in the OASIS (Organization for the Advancement of Structured Information Standards) organization. Those efforts resulted in a standard definition called WSDM (Web Services Distributed Management). The latter is made of two specifications called MUWS (Management Using Web Services) [22] and MOWS (Management Of Web Services) [23]. The first specification

(MUWS) could be a standard way to implement the Effector and Sensor manageability interfaces of our proposed architecture so that autonomic managers use Web Services technologies to control managed resources. The adoption of this specification improves interoperability between distributed heterogeneous managed resources and autonomic managers thanks to standardized Web Services technologies usage. MUWS provides a standard and flexible way to specify manageability capabilities exposed by managed resources manageability interfaces. Those manageability capabilities are extensible so it becomes possible to have a full coverage of the self-management objectives in our architecture.

Thanks to the different interactions (vertical and horizontal) that could take place between different components of the proposed architecture, we can provide mobile users with an end-to-end QoS and security guarantee while using NGN services.

4. QoS and security negotiation framework for mobile users

The objective of this section is to describe the functioning of the negotiation provided by SLNP in the NGN environments. We start by introducing the various messages used by SLNP in a service level negotiation, before describing the structure of the SLS which enables a simultaneous negotiation of QoS and security. After that, we detail the user profile on which the negotiation process is based. This allows the automation of the negotiation in order to allow the adaptation of the service offer following any occurring changes. Then, we show the composition of an SLNP entity before describing the global SLNP negotiation process.

4.1 SLNP negotiation messages

The terminology adopted for the use of our protocol (SLNP) in conformance with the autonomous management architecture described in section 3 is the following (Figure 4):

- *SHE (SLNP HAM Entity):* a HAM entity that supports SLNP.
- *SHI (SLNP HAM Initiator):* a SHE that initiates the negotiation process.
- *SHR (SLNP HAM Responder):* the last SHE on the negotiation path.
- *SHF (SLNP HAM Forwarder):* any SHE between a SHI and a SHR and participating to the negotiation process.

SLNP was defined to guarantee an end-to-end service level negotiation between the managers of the different domains implied in a service offer [15]. This negotiation is performed thanks to the exchange of negotiation messages which enable the establishment, the modification and the termination of a service level.

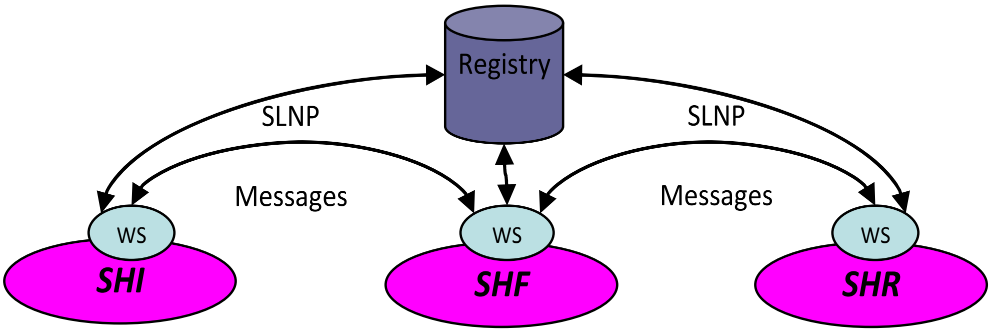


Figure 4. SLNP terminology

Since SLNP uses Web Services, the negotiation is based on the exchange of SOAP (Simple Object Access Protocol) messages which can have one of the following types.

- *Negotiate:* generated by the initial entity (SHI) toward the final entity (SHR), this message allows requesting the establishment of a service level (SLS) through the specification of the negotiated parameters and their values.

- *Revision:* sent by the SHR toward the SHI in order to propose an alternative to the requested SLS.

- *Modify:* generated by the SHI toward the SHR, this message enables the SHI to request the modification of an established SLS. This request can be motivated by changes in the service requirements or following the reception of a notification (Notify) informing about changes in the network capabilities.

- *Notify:* sent by any intermediate entity (SHF) or by the SHR, directly toward the SHI to request improvement or degradation of the established SLS.

- *Release:* sent by the SHI toward the SHR in order to release an established SLS.

- *Response:* generally sent by the SHR toward the SHI in order to answer a request of type Negotiate, Modify or Release. It can also be sent by the SHI toward the SHR in order to answer an alternative (Revision) or a notification (Notify).

Each of the above presented messages contains an SLS element that allows specifying the negotiated SLS. The structure of this SLS is described in the following section.

4.2 SLS definition for SLNP negotiation

Since SLNP is based on the use of Web Services, the structure of the defined messages and the composition of the specified SLS are described using XML schemas. Figure 5 provides a graphical representation of the SLS XML schema that enables the negotiation of both security and QoS. In this figure, optional parameters are presented in dotted boxes.

We classify SLS parameters defined for SLNP negotiation into three catergories: general parameters, parameters related to QoS and those related to security.

4.2.1 General parameters

These parameters, which are common to security and QoS, are included in the “sls” element and contain:

- *SLS identifier:* a mandatory element which enables the identification of the negotiated SLS and guarantees unicity when defined by SHI.
- *Flow identification:* this mandatory element permits the identification of the traffic concerned by the negotiation through IP addresses, ports, protocol id, etc.
- *Negotiation parameters:* this optional element contains two parameters; the negotiation mode which can be predefined or non-predefined, and the negotiation interval representing the duration after which the SLS can be renegotiated.
- *Reliability:* is an optional element which includes two parameters: the Mean Down Time (MDT) and the Mean Time To Repair (MTTR).

4.2.2 QoS parameters

The parameters related to QoS are gathered into a single element: “qosParameters” (Figure 5). This element is mandatory because the negotiation must at least relate to QoS, and includes five elements:

- *Scope:* this mandatory element allows specifying the geographic boundaries within which QoS must be guaranteed. It is composed of ingress and egress.
- *Service schedule:* a mandatory element which specifies the time during which the negotiated SLS must be guaranteed. This time can be immediate, differed or periodic.
- *Performance guarantee:* this mandatory element contains the parameters generally used to specify the QoS. These parameters are: delay, jitter, loss rate and bandwidth.
- *Traffic conformity:* is an optional element that enables distinguishing the in-profile trafic and the out-of-profile trafic using a set of indicators like packet size, Commited Information Rate (CIR) Peak Information Rate (PIR) etc.
- *Excess treatment:* this optional element specifies the treatment applied by the network to out-of-profile packets. This treatment may be: drop, shape or mark.

General parameters and QoS parameters have been mainly inspired from the attributes defined by the protocols specified for QoS negotiation (section 2.2.2).

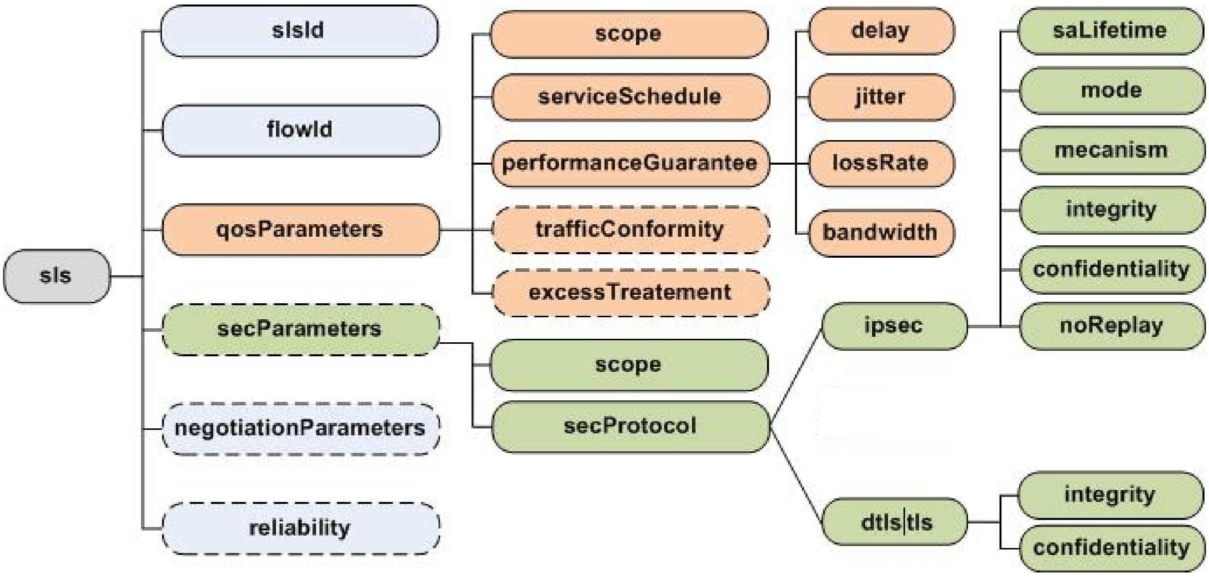


Figure 5. XML schema of the defined SLS [24]

4.2.3 Security parameters

As for the parameters related to security, they are included in an element called “secParameters”. This optional element is constituted of two mandatory sub-element: “scope” and “securityPotocol”.

- *Scope:* this mandatory element specifies the geographic boundaries within which the required security services must be guaranteed. It has the same composition as the scope contained in QoS parameters, but it can be different from it. Indeed, in a QoS context, scope ingress and egress indicates the two communication ends (e.g. two terminals). Whereas from a security point of view, they can constitute the security association peers (e.g. security gateways to which terminals are connected to).
- *Security protocol:* this element enables selecting the protocol to use: IPsec or TLS/DTLS. For each one of these two security types, there is a set of parameters which are negotiated in order to specify the security level.
- *IPSec parameters:* the “ipsec” element contain six sub-elements: “saLifetime”, “mechanism”, “mode”, “integrity”, “confidentiality” and “noReplay”. These elements allow specifying the amount of the protected data, the used mecanism and mode, the offered security services and the used algorithms.
- *TLS/DTLS parameters:* the “tls” element includes only two sub-elements: “integrity” and “confidentiality” which permit to select the security services to offer and the algorithms to use.

The most used security protocols are: IPSec (IP Security Protocol) [25], which secures any type of traffic at the "Network" layer, and TLS (Transport Layer Security) [26] and

DTLS (Datagram Transport Layer Security) [27] that allows introducing security at the "Transport" layer for respectively TCP and UDP traffics. Thus, security parameters to negotiate with SLNP were defined on the basis of these protocols. Indeed, the negotiated IPSec parameters are used to configure a security association that provides security services to the concerned traffic. As for TLS/DTLS negotiated parameters, they specify the security characterizing the session to establish between the communication's ends. We note that the security parameters related to TLS and DTLS are the same because these two protocols offer the same security guarantees using the same mechanisms.

The association of security and QoS in a service level is very important since it allows taking into account the impact of security on QoS. Indeed, when security services are provided, they can have an impact on the communication QoS. This impact depends on several conditions, but it must be taken into account when negotiating a SLS (section 4.5.1).

4.3 User profile for SLNP negotiation

In the context of a SLNP negotiation, the user profile is used to store information about the communication environment for which the service level (SLS) must be negotiated. This information is related to terminal, application, access network, and user preferences (Figure 6), and helps to establish, modify and release the SLS. In this section we detail selected information that will constitute the user profile allowing the automation and the optimization of the negotiation in autonomous NGN environments.

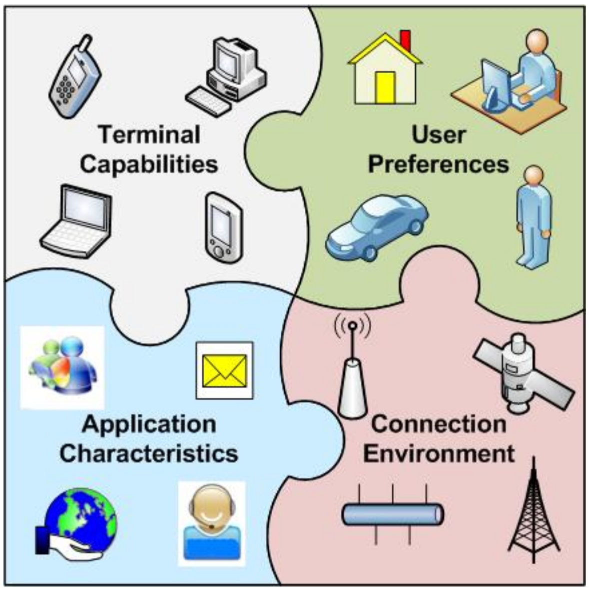
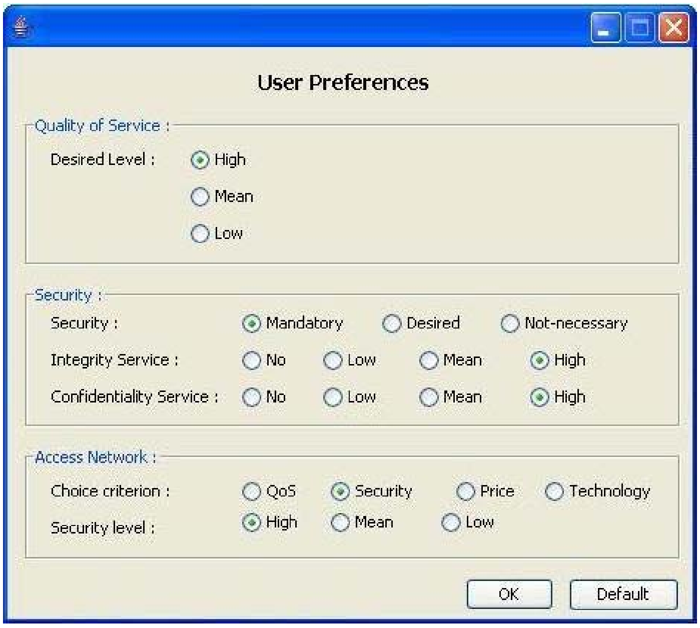


Figure 6. Profile structure

4.3.1 User preferences

User preferences are divided into three categories: QoS, Security and Access network. Regarding QoS, preferences are expressed by the desired level: High, Medium or Low. When High or Medium is selected, the negotiated QoS level is at least equal to the QoS level required for the normal functioning of the application (Minimal). For security, user must specify if security is Mandatory, Desired or Not-necessary. In the two first cases, this user should select the needed security services (Authentication, Integrity, Confidentiality and No-replay) and the level of every selected service (High, Medium or Low). If security is Mandatory, then established security level must satisfy user needs. If security is Desired, then security level will, at best, correspond to that specified by the user. Regarding access networks, user preferences are expressed by selecting a criterion for access network choice such as technology, QoS, Security or Cost. Then, user will specify how this criterion is used in network choice. For example, if access technology is the selected criterion, then user will classify preferred access technologies that can coexist. Thus, if a priority network is detected, then a handover can be executed and may lead to changes in the established service level. A Graphical User Interface (GUI) is used to collect user preferences (Figure 7).



User Preferences

Quality of Service :

Desired Level : ● High ○ Mean ○ Low

Security :

Security : ● Mandatory ○ Desired ○ Not-necessary

Integrity Service : ○ No ○ Low ○ Mean ● High

Confidentiality Service : ○ No ○ Low ○ Mean ● High

Access Network :

Choice criterion : ○ QoS ● Security ○ Price ○ Technology

Security level : ● High ○ Mean ○ Low

Figure 7. GUI for user preferences

4.3.2 Application characteristics

Essentially composed of the Name and the Type of the application, these parameters provide the entity initiating the negotiation with information about the minimal needed QoS level. Indeed, a mapping table provides QoS requirements according to the application type. Since an application may have its own security mechanisms such as Web Service Security (WSS) [28], Security information must be among the application characteristics.

4.3.3 Terminal capabilities

Among these capabilities, we find parameters such as Screen size and Supported codec (video and audio) that provide indications on the required QoS. Moreover, these characteristics include Performance parameters like CPU and Memory that help estimating security impact on QoS. These characteristics also contain Security protocols and Cryptographic algorithms which are supported by the terminal. This helps on defining security parameters to negotiate with SLNP.

4.3.4 Network characteristics

An access network is generally characterized by: an Identifier, an Access technology (PSTN, ADSL, GPRS, UMTS, Wi-Fi, WiMax, Bluetooth, etc.), a Cost, Qos and Security parameters. QoS parameters include Bandwidth, Latency, Jitter and Loss rate. While security parameters specify the used Security protocol such as WEP (Wired Equivalent Privacy), WPA (WiFi Protected Access) or WPA2 which enables securing Wi-Fi networks. With user preferences in terms of access network, these parameters permit the selection of the access network the better corresponding to the user context. The information about available networks can be collected using the 802.21 [29].

User preferences, application characteristics, terminal capabilities and characteristics of the available access networks are collected by the user terminal on which the communicating application is executed. Then, these parameters are transmitted by this terminal to the entity which will initiate the SLNP negotiation (SHI).

4.4 SLNP entity composition

A negotiation entity may be involved in several negotiation processes and can play different roles. Indeed, an SLNP entity can play the role of SHI in a negotiation process, and initiates a service level negotiation. It can also play the role of SHF or SHR solicited by another entity in another negotiation process. Thus, to ensure the good functioning of SLNP, a negotiation entity must be composed of a Negotiation Client Application (NCA) to initiate any type of negotiation process and a Negotiation Web Service (NWS) containing the different operations in order to treat the various requests that an entity could receive. A negotiation entity must also contain, first, a Mapping to define the negotiated parameters on the basis of the user profile and, second, a Negotiation Decision Point (MNDP) to make negotiation decisions. In the remainder of this section, we describe these three components: NCA, NWS and MNDP.

4.4.1 Negotiation Client Application (NCA)

An initial entity (SHI) must contain an NCA to establish, modify or terminate a SLS by invoking the NWS of the next entity with the appropriate message (Negotiate, Modify or Release). The client application of an SHI should also respond to a proposed alternative by invoking the next NWS with a response message (Response). An intermediate or a final entity (SHF or SHR) must contain an NCA which is used to generate a notification message (Notify) and directly invoke the NWS of the initial entity (SHI). Thus, an NCA must be able to invoke any operation contained in the NWS using the appropriate message.

4.4.2 Negotiation Web Service (NWS)

An initial entity (SHI) must contain a NWS which will receive and respond to notifications (Notify) transmitted by other entities to provide information on possible changes in resources availability or non-conformance to an already established SLS. However, an intermediate or final entity (SHF or SHR) must contain a NWS to receive various requests (Negotiate, Modify and Release), treat them and return a response (Response) or an alternative (Revision) to the calling entity. The returned message (Response or Revision) can be generated by the entity itself or can be the retransmission of the message returned by the next entity of the negotiation path. The NWS of the SHF or SHR must also enable it to receive a response (Response) to its already proposed alternative. Thus, the NWS of an SLNP entity must be composed of five operations. The NWS is defined through its WSDL (Web Services Description Language) description represented in Figure 8. This definition covers the various operations and the types of input and output messages associated to each operation.

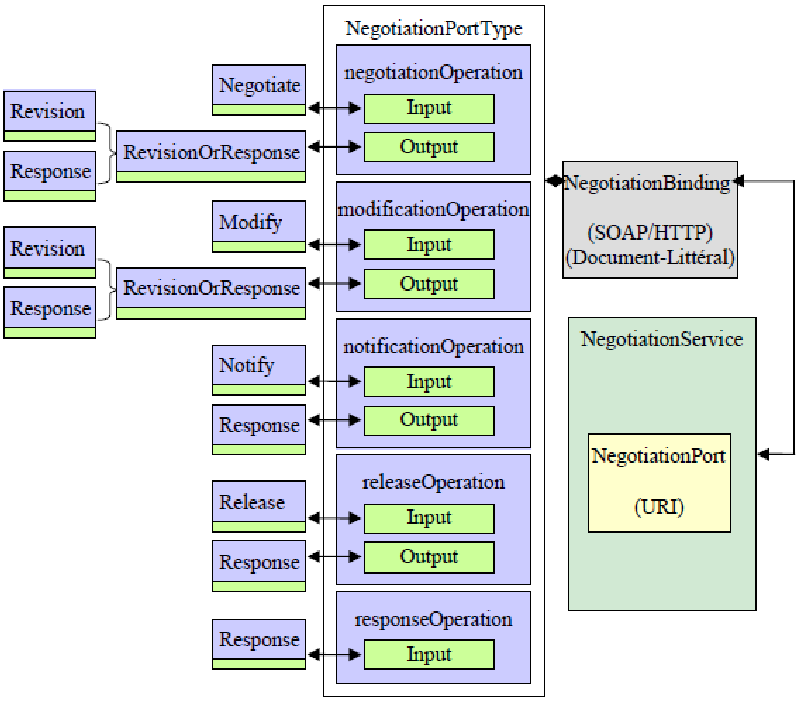


Figure 8. WSDL representation of the NWS [8]

4.4.3 Mapping and Negotiation Decision Point (MNDP)

The MNDP is responsible for making decisions related to the negotiation like SLS establishment, modification and release. These decisions are based on user profile parameters and changes that may occur (Figure 9). Then, when negotiation process should be started, the MNDP must provide the client application with the SLS to negotiate. This SLS is defined according to user profile parameters and will be included in the message needed to initiate the negotiation process. The algorithm describing the treatments included in the MNDP is provided in section 4.6.1.

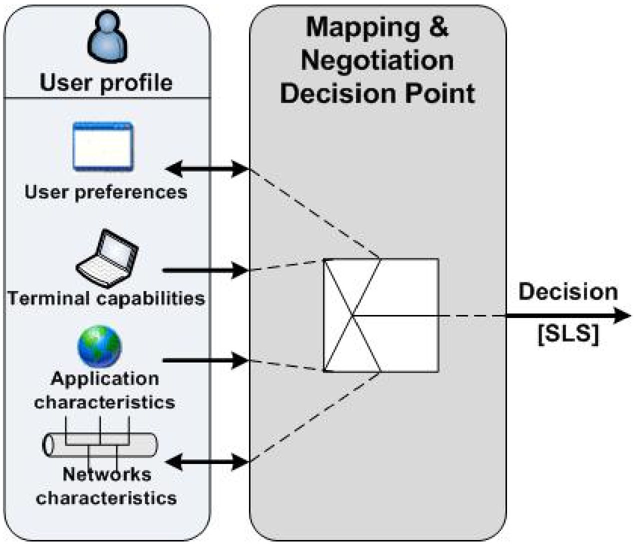


Figure 9. Interactions of the MNDP

The above described components (Figure 8 and 9) form what we call a negotiation layer of an SHE. This layer have to interact with a storage component: the SLS Registry (SR) used to record established SLS.

4.5 SLNP negotiation process

In this section, we describe the functioning of the dynamic SLNP negotiation through a scenario corresponding to a communication between a mobile user (USER) and a fixed server (SERVER) shown in Figure 10. The main steps which can be included in a negotiation scenario are: the SLS establishment, the modification and the release of this SLS.

4.5.1 SLS establishment

In the considered example (Figure 10), the SHI have to negotiate a SLS for the communication between USER and SERVER. To satisfy adequately the needs of this communication, the SLS to negotiate is defined on the basis of the profile of the mobile user (USER). Thus, the first step is to collect profile parameters by the terminal. Then, the latter transmits them to its autonomic domain manager (SHI) that will handle the negotiation. Once the user profile is received, the SHI initiates the SLNP negotiation of a SLS which meets the needs of USER. For this, the parameters of the required SLS are specified thanks to a mapping process (section 4.6.1) performed by the MNDP. Then, the QoS local offer is added to this SLS after interacting with the different LAMs involved in the communication. Indeed, the interactions with LAMs enable an SHE to obtain the needed information about the managed domain (e.g. the network capabilities, resources availabilities, etc.) and, then, treat properly the received requests. After that, the desired SLS is negotiated with the autonomic managers of the crossed domains (SHF and SHR) by transmitting a Negotiate message to the SHF (Figure 11). In order to process the Negotiate message (updating it following its QoS local offer), the SHF must interact with its LAM that provides it with information on resources availability and requests admissibility. After that, the Negotiate message is forwarded to the SHR which has to make a decision concerning the negotiated SLS. This decision depends on the requested level and the offered level, and can be acceptance, rejection or alternative proposition. When negotiation entities (SHI, SHF and SHR) agree on the negotiated parameters, SLS is established and recorded in SLS registries. After that, QoS will be guaranteed by configuring the concerned entities (Edge Routers), and security will be offered at the network level using IPsec or at the transport layer using TLS or DTLS. Security configuration is done by transmitting the needed information to the concerned entities. When transport mode of IPsec is used, this will be done by transmitting security policies to the endpoints of the communication (USER and SERVER) in order to indicate the traffic to secure and to specify how to secure it. When the first packet is sent, a security association will be automatically set up by a security association management protocol (e.g. IKEv2). Within TLS, the negotiated parameters will be passed to the TLS implementation in order to specify cryptographic suites which can be used by the TLS Handshake Protocol in order to configure the session security parameters.

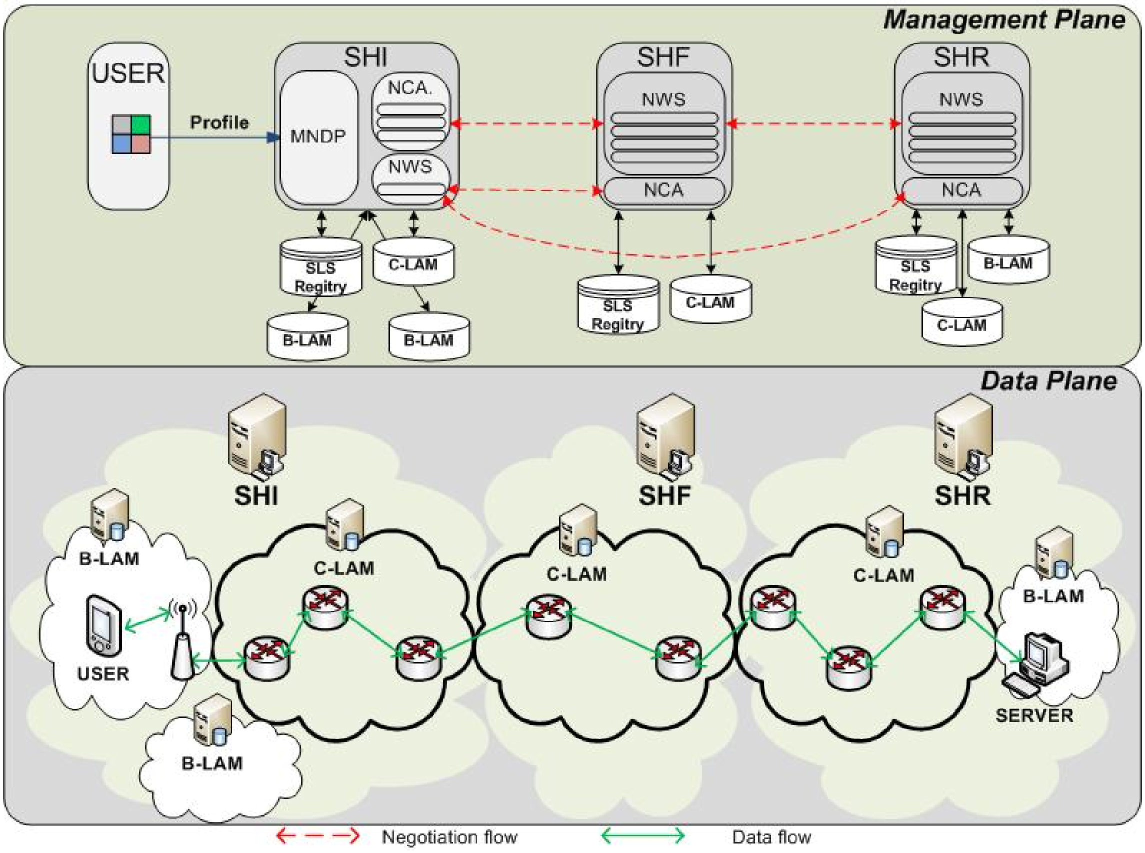


Figure 10. SLNP negotiation process in NGN environments

When security services are provided to a communication, the computing operations and the protocol overhead are increased [30]. The protocol overhead is due to the fields added to the original packets in order to achieve security. The operations include initialization of security mechanisms as well as cryptographic operations. These operations increase resource consumption, such as memory, CPU and bandwidth. The security impact on QoS is more or less important according to the set of the selected services and to the set of algorithms used to provide these services. For example, integrity has an impact on the bandwidth usage since authentication data are added to original packets. Similarly, confidentiality requires cryptographic operations such as encoding and decoding. This can introduce a processing delay in addition to the normal packet transit delay.

Since security impact on QoS could, in some cases, prevent normal communication functioning, it is very important to consider it when negotiating a SLS. In fact, this impact must be expressed in terms of delay and bandwidth, and must be considered during service level negotiation. Indeed, when an SHE processes a message by updating the parameters values (e.g. adding the delay of transit through its network in order to calculate the end-to-end offered delay), it must take into account the delay involved by cryptographic operations such as the MAC (Message Authentication Code) calculation, encryption and decryption. Similarly, when the SHI initiates a negotiation and specifies the bandwidth value needed to exchange data, it must take into consideration the bandwidth consumed by security services.

4.5.2 SLS modification

The SLS modification could occur following two kinds of changes: changes in the user profile or evolution of resources availability.

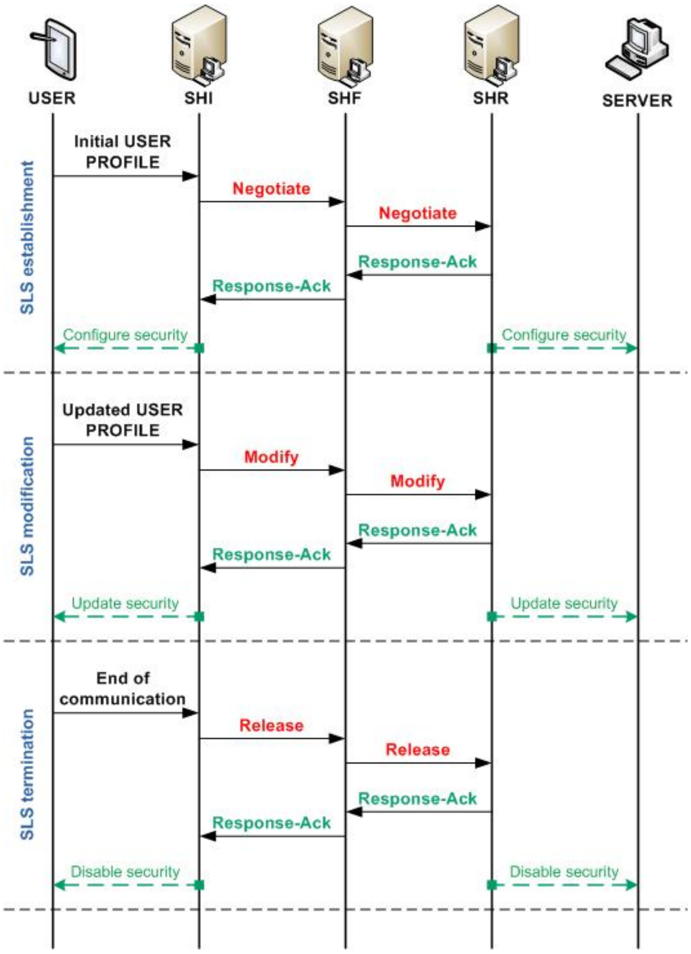


Figure.11. MSC of a negotiation scenario

- Changes in the user profile can result from the evolution of user preferences or from the modification of the network characteristics which can be caused by the execution of a handover while the mobile user is moving. Indeed, the mobility of users can cause the modification or the cancellation of an already established SLS. It is an intra-domain mobility consisting in a handover between two access networks belonging to the same autonomous system, and managed by the same HAM.

- Changes in the resources availability within one of the implied domains must cause the transmission of a Notify message to the SHI. Following the reception of this notification, the SHI can proceed to the modification or the cancellation of the already established SLS. Thus, the SLNP protocol allows the involved HAMs to inform the SHI about the perceived QoS.

As we said before, an established SLS can be modified following SHI request. Whatever the cause, a modification process is initiated through the transmission of a Modify message by the SHI and requires the definition of a new SLS at the MNDP of this SHI. In the provided example (Figure 10), we note that SLS modification is caused by changes in the user profile following a handover between two access networks. In fact, when the user profile, located in the SHI, is updated, the MNDP performs a new mapping which leads to the definition of a new SLS that will be different from the already established one. This means that the needs have changed and that the already established SLS should be modified. The SLS modification process is very similar to that of the establishment. In fact, a Modify request is sent to the SHR via the SHF (Figure 9), and this request can be accepted, rejected or an alternative is proposed. In the case of an acceptation, the SLS is modified and all the LAMs involved in the corresponding service offer must be informed. This allows them to arrange the resources allocation: free or allocate resources if needed. If security parameters are changed, then a new configuration will be done by transmitting the needed information to the concerned entities.

In the case of an inter-domain mobility, the already established SLS must be released and a new one must be established. The establishment of the new SLS will be initiated by another HAM which does not dispose of the already collected user profile. Thus, the inter-domain mobility is not addressed in this paper.

4.5.3 SLS termination

The termination of a SLS occurs when a communication having an immediate or a differed service schedule is over, or when the service schedule of a periodic reservation expires.

An established SLS is terminated following SHI request. In fact, this entity can initiate the termination process by sending a Release message to the SHR via the SHF. The SLS contained in this message is only used for specifying the concerned SLS (sls Id). In the provided example, the SLS termination is initiated following the end of the communication (Figure 11). The SLS termination process must always happen in one round: one request (Release) and one response (Response). If the indicated SLS is already existing, then the response will be positive (Response-Ack) and the SLS will be terminated. Otherwise, a negative response will be returned (Response-Nack) while specifying, in the “reason” field of the Response message that the indicated SLS does not exist. When a SLS is released, all the LAMs involved in the corresponding service offer must be informed, in order to allow them to free the resources allocated for the released SLS. In addition, if security is included in the terminated SLS, then ends of the secured communication (USER and SERVER) must be informed in order to disable the configured security level.

4.6 SLNP negotiation operations

In this section, we detail the treatments included in the different components of the negotiation layer described above.

4.6.1 MNDP treatments

As mentioned before, in order to optimize and automate the SLS negotiation, the MNDP ensures many tasks. In the following, we detail the various treatments (actions and tests) included in the MNDP (Figure 12).

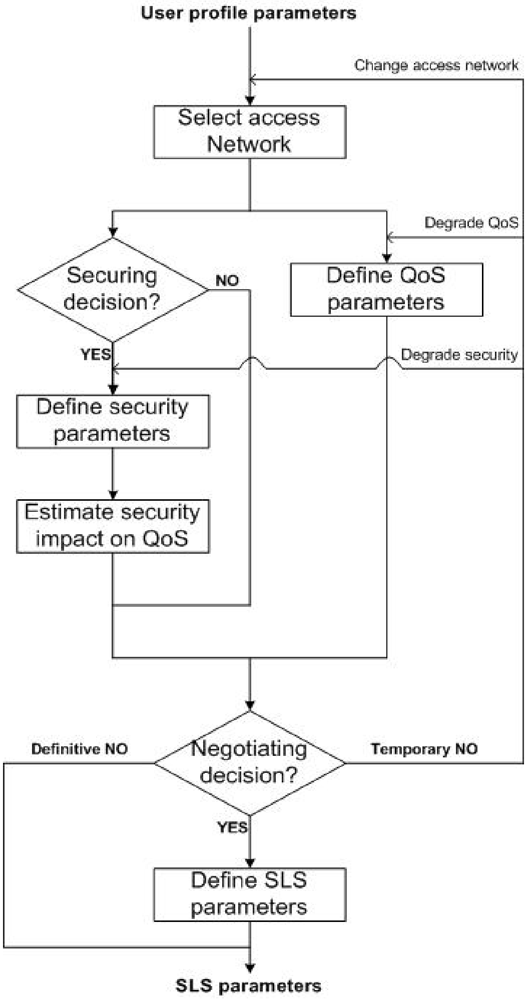


Figure.12. MNDP processing diagram [24]

- *Select access* network: The choice of access network is based on: available networks characteristics and user preferences for access networks. The identifier of selected network is returned to the terminal in order to enable its connection to this network.

- *Define QoS parameters:* Application name and type, screen size and supported codecs help to define the minimal QoS level. Then, user preferences in terms of QoS are used to specify QoS parameters to negotiate and a boolean value "qos degradable" indicating whether the requested QoS level can be degraded.

- *Securing decision:* This decision is made on the basis of security information about: access network, application and user preferences. Indeed, if required security level (user preferences) is neither ensured by the access network nor enabled by the application, then the decision is to secure communication at the network or the transport layer.

- *Define security parameters:* When security services must be provided, security parameters are defined according to user preferences (selected security services and desired level for each service) and terminal characteristics (supported protocols and algorithms). Then, these parameters are transmitted with a boolean value "security degradable" which indicates whether security can be degraded.

- *Estimate security impact on QoS:* When security is required, its impact on QoS is estimated based on security parameters and terminal performances. Indeed, protocols' overheads depend on security mechanisms and services, while introduced latency depends on these same factors, but also on terminal performances.

- *Negotiation decision:* This decision depends on: required QoS, estimated impact, and access network quality. First, the previously defined QoS level is adjusted taking into account security impact. Then, if the QoS provided by access network can not ensure the QoS required by the communication, then the negotiation can not be started. For example, it is not necessary to negotiate a SLS for a communication requiring a bandwidth which is superior to that guaranteed by the access network. In this case, if QoS level and/or security level can be degraded, then a degradation request is sent to 'define QoS parameters' and/or to 'define security parameters' module. The type of parameters to degrade (QoS, security, or both) depends only on the strategies implemented by MNDP. This mechanism provides an internal negotiation, which enables avoiding the loss of time that can be caused by rounds of negotiations between all the implied SHE. Degradation request may be sent one or more times until it becomes possible to start the negotiation or until the requested SLS can not be further degraded. In the latter case, if other networks are available, a handover is executed and the negotiation is retried.

4.6.2 NCA treatments

In the context of a SLS establishment, the treatments included in the NCA located at the SHI are represented by the diagram of Figure 13.

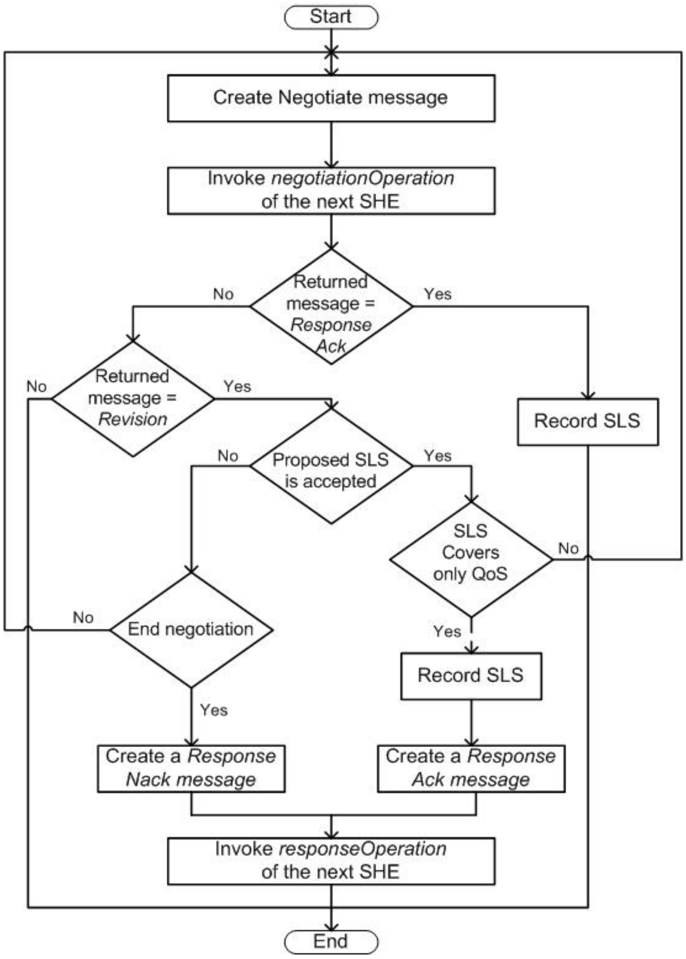

Start
Create Negotiate message
Invoke negotiationOperation of the next SHE
Returned message = Response Ack
No
Yes
Returned message = Revision
No
Yes
Record SLS
Proposed SLS is accepted
No
Yes
SLS Covers only QoS
No
End negotiation
No
Yes
Record SLS
Yes
Create a Response Nack message
Create a Response Ack message
Invoke responseOperation of the next SHE
End

Figure 13. NCA processing - establishment

From this diagram, we note that the establishment of a SLS starts by the constitution of a Negotiate message using the SLS element generated by the MNDP. Then, this message is used to invoke the negotiation operation (NWS) of the next SHE on the negotiation path. According to the type of the returned message, following the invocation, there are five different behaviors. First, if the requested SLS is accepted by the whole network (returned message = Response-Ack), then SHI should record the SLS, and the negotiation process is over. The second scenario is the rejection of the SHI request (returned message = Response-Nack) which does not require any treatment and leads directly to the end of the negotiation. It is also possible to return a Revision message to the SHI. This message allows proposing an alternative to the SLS requested by that SHI. When the proposed alternative can not be accepted, the SHI should reject it by sending a Response-Nack message or by starting a second round of negotiation. Otherwise (the proposed alternative can be accepted), there is two possibilities. Indeed, if the negotiation is limited to QoS, then SHI have to send a Response-Ack message after recording the accepted SLS. However, if security is included in the negotiated SLS, the proposed alternative can not be accepted because the contained QoS level is determined on the basis of security settings initially requested by the SHI. Hence, a second round is required in order to ensure that the contained QoS matches with the specified security.

4.6.3 NWS treatments

Treatments contained in the various operations composing the NWS enables an SHE to handle different requests that can be received. To have an idea on these treatments, we consider the operation which is invoked while establishing a SLS (negotiation operation). The invocation of this operation is always initiated by the SHI, and lead to a recursive call to the negotiation operations of all SHE involved in the negotiation. To optimize the SLNP implementation, the algorithm of the negotiation operation treatments, shown in Figure 14, has been defined in a manner that enables covering the behavior of both intermediate and final entities. In the case of an intermediate entity (SHF), the SLS element is extracted from the received message (Negotiate). Then, the needed modifications are performed before reconstructing the Negotiate message and forwarding it to the next SHE on the negotiation path. After that, the returned message, following that invocation, is forwarded to the previous entity after recording locally the SLS in the registry reserved for this purpose, when an agreement is reached (Response-Ack). In the case of a final entity (SHR), after modifying the received SLS according to local offer, a comparison between the end-to-end offer and the initial request is performed in order to decide if the SLS requested by the SHI can be accepted. If the SLS requested can be satisfied, then the SHR may return a Response-Ack message after recording locally the established SLS. Otherwise, the SHR may propose an alternative using a Revision message, or may terminate the negotiation by rejecting the SHI request through a Response-Nack message.

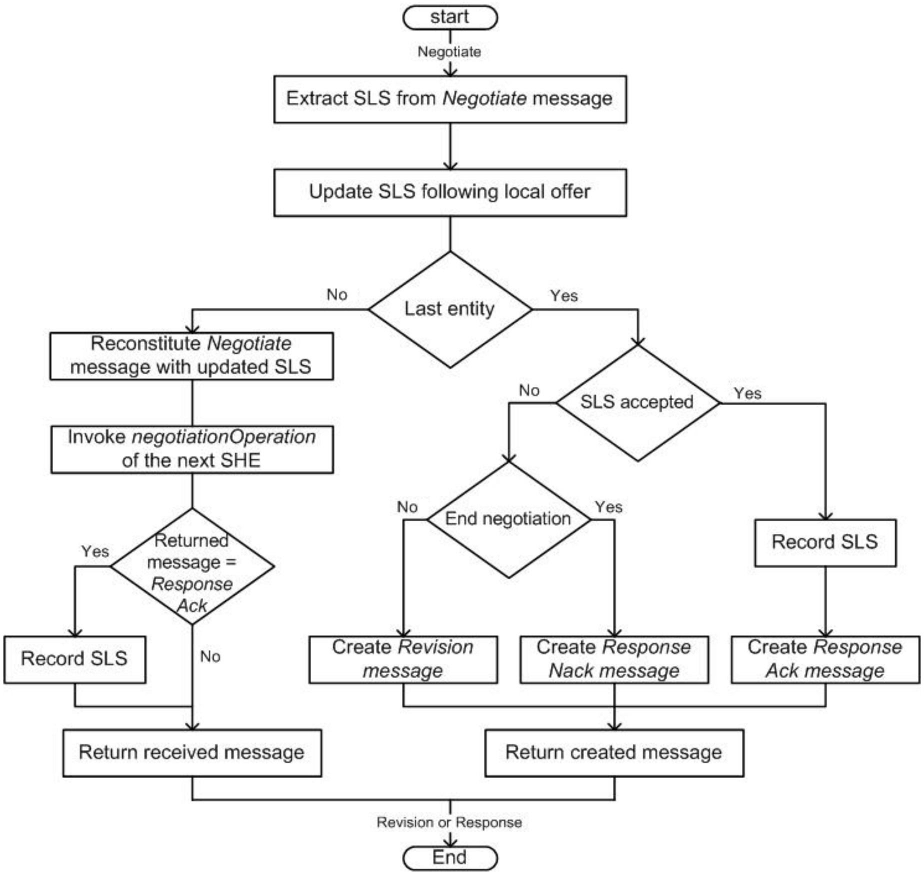

start
Negotiate
Extract SLS from Negotiate message
Update SLS following local offer
No
Last entity
Yes
Reconstitute Negotiate message with updated SLS
No
SLS accepted
Yes
Invoke negotiationOperation of the next SHE
No
End negotiation
Yes
Record SLS
Yes
Returned message = Response Ack
Record SLS
No
Create Revision message
Create Response Nack message
Create Response Ack message
Return received message
Return created message
Revision or Response
End

Figure 14. NWS processing - negotiation operation

5. Tests and results

5.1 SLNP performance tests

Once the implementation of SLNP negotiation is accomplished, we conduct a set of tests in order to evaluate the performance of this negotiation.

5.1.1 Platform description

To evaluate the SLNP negotiation performances, we use the experimental platform shown in Figure 15. On this platform we have deployed three SHE that implement SLNP. In the implementation of each SHE, we choose to use the Apache-Tomcat server [31] on which the NWS of each entity is deployed. As implementation of the SOAP protocol, we opted for the Apache implementation called AXIS [32] which is easily deployed on the Tomcat server. To manage the databases used for SLS storage, we used MySQL [33]. Implementation of the different negotiation layer components has been realized using the Java programming language [34]. These choices do not imply that each SHE must use the same platform and the same programming language. Instead, SLNP protocol uses web services to enable interoperability between different autonomic managers which want to negotiate an end-to-end service level. To accomplish local tests, the three SHE have been deployed on a Dell Precision PWS670 system equipped with an Intel Xeon 3.2 GHZ processor and a 2GB RAM.

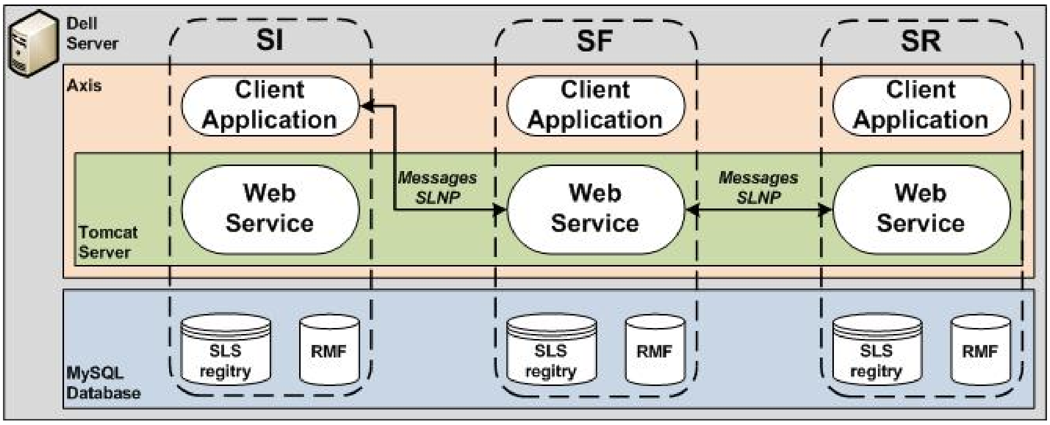


Figure 15. Platform used in SLNP performances tests

5.1.2 SLNP negotiation performance

After testing the good functioning of the negotiation, we evaluate the negotiation performances (negotiation time and message size) for two negotiation scenarios: negotiation of only QoS and negotiation of both QoS and security.

- *Negotiation time:* The evaluation of the negotiation time is based on one-round scenario in order to compare the time needed for QoS negotiation to that required for a negotiation of both QoS and security. We note that the measured time corresponds to the time between the moment when the SHI starts the construction of a Negotiate message and the moment of the reception of a Response message. We also note that the mean negotiation time is calculated based on a sample of 1000 measurements. However, in Figure 16, we show a sample of only 100 measurements in order to provide a better visualization of the tests results. Under these conditions, we obtained a mean time equal to 60.08 ms for a round of QoS negotiation (red curve in Figure 16). However, in the case of a joint negotiation of QoS and security, we got a mean time of 62.61 ms for a single round (blue curve in Figure 16). From these measurements, we conclude that introducing security in the negotiated SLS increases slightly (4.21%) the mean time of a negotiation round. This can be explained by the fact that the negotiation treatments represent generally a small fraction of the negotiation time. In fact, most of the negotiation time corresponds to the time of Web Services invocations.

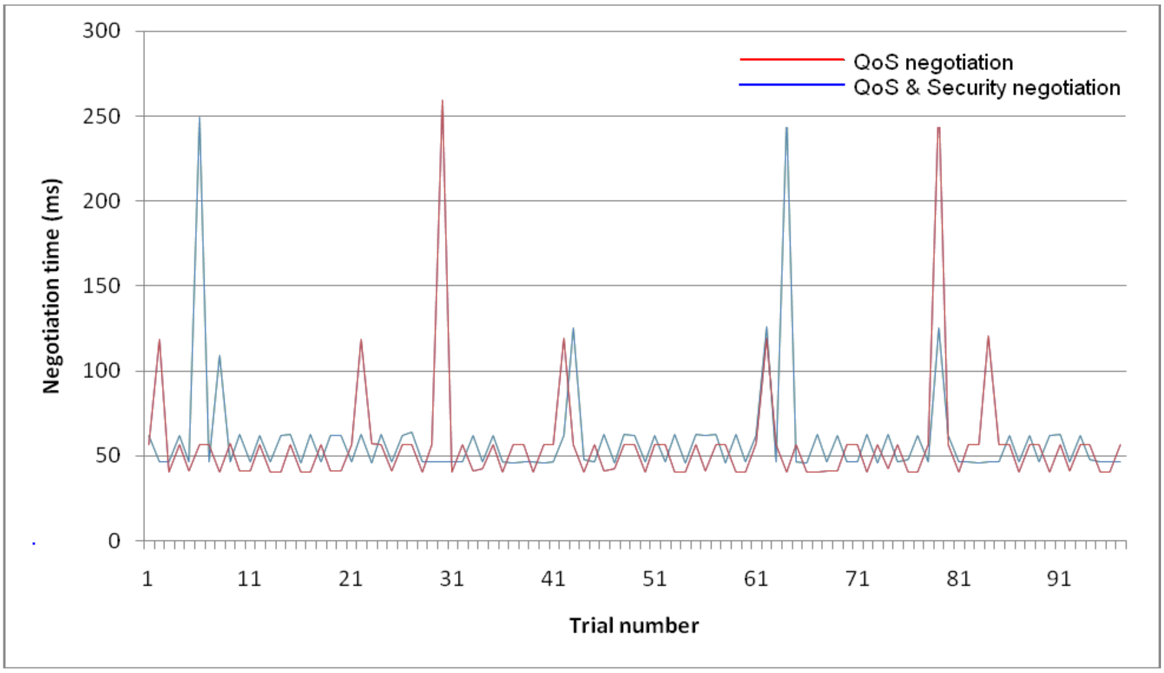


Figure 16. Negotiation time depending on the negotiated parameters

- *Message size:* To measure the size of messages exchanged between the negotiation entities of the test platform, we used the SOAP Monitor tool of Apache which allowed us to capture and store these messages in order to measure theirs sizes. Since all the SLNP messages have almost the same structure, the size of these messages will depend mainly on the parameters included in the negotiated SLS. Thus, we measure the average size of a Negotiate message for two types of negotiation: the negotiation of only QoS and the joint negotiation of QoS and security. The measurement results show that the average size of a Negotiate message increases when security parameters are introduced in the negotiated SLS (Figure 17). Indeed, this size goes from 3749 bytes to 4894 bytes, which represents an increase of about one third the size of the original message (30.54%). This important increase is obviously due to the addition of XML elements corresponding to the negotiated security parameters.

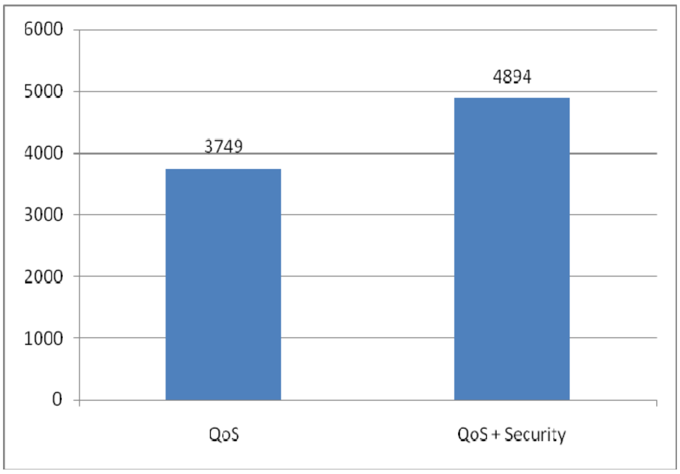


Figure 17. Message size depending on the negotiated parameters

5.2 Impact evaluation tests

In order to take into account the security impact on QoS while performing a SLNP negotiation, negotiation entities must have an estimation of this impact. Thus, we have conducted some tests that we present in this section.

5.2.1 Platform description

The tests enabling the impact estimation are performed on a platform composed of two IBM systems having the same performances; a Pentium IV 2.4 GHz CPU and a 256 MB RAM. The performed tests consist in establishing secure communications between the two platform systems through the wired network connecting them (Figure 18). The concerned security protocols are IPSec, TLS and DTLS. Thus implementations of these protocols have been established on both the two platform systems (Openswan [35] for IPSec and OpenSSL [36] for TLS/DTLS). The tools used to achieve our tests are: Sjitter [37], Iperf [38] and Hping [39].

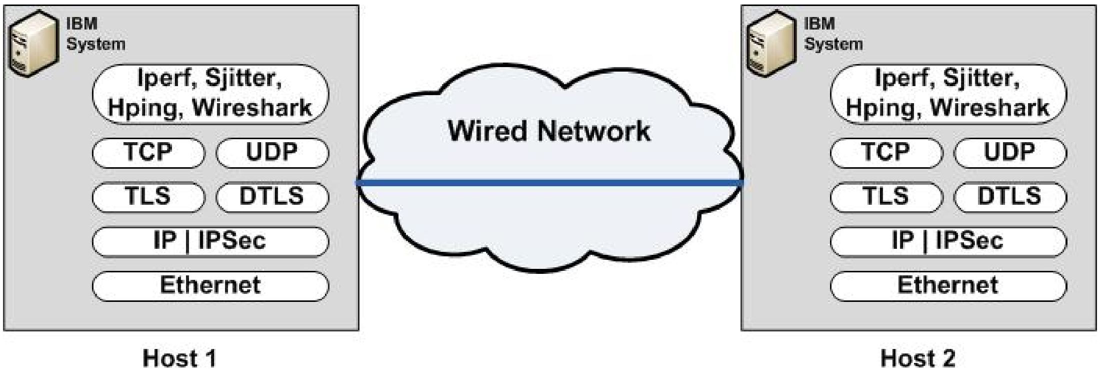


Figure.18. Platform used in impact estimation tests

Since several factors can cause the degradation of the transmission QoS like the network overload and CPU use rate, the experimental conditions were extremely controlled. Indeed the two systems were running only the application that ensures the data transmission. In addition, the wired network connecting the two systems was over-provisioned (the bandwidth in the network is greater than the required throughput) and there is no other data transmitted on this network.

5.2.2 Impact evaluation

The impact evaluation is achieved through two sets of measurements. In the first set, we tried to evaluate the impact of a security service on the QoS parameters varying the algorithm used in the security service offer. The obtained results let us conclude that the choice of the algorithm used in providing a security service (integrity or confidentiality) has practically no influence on the impact measurements. Thus, the second set of measurements aims to evaluate the impact depending on employed security protocols and provided security services.

The second set of measurements concerns two types of traffic: UDP and TCP. Thus, these measurements were performed following security policies summarized in Table.1. P0, P1, P2, P3 and P5 represent the policies applied for tests related to the UDP traffic, while P0, P1, P2, P4 and P6 concern tests corresponding to the TCP traffic.

Table 1. Security policies used in impact evaluation

Policy	Characteristics
P0	No security
P1	IPSec, AH, Integrity=hmac-sha1-96
P2	IPSec, ESP, Integrity =hmac-sha1-96, Confidentiality=aes-cbc
P3	DTLS, Integrity =sha1, Confidentiality =aes
P4	TLS, Integrity =sha1, Confidentiality =aes
P5	P2 + P3
P6	P2 + P4

- *UDP traffic:* Figure 19 shows security impact on delay, jitter and bandwidth for UDP traffic. We note clearly that security services (integrity and confidentiality), provided with IPSec and / or DTLS, increase the delay and the jitter of an UDP transmission and require additional bandwidth. We note also that the impact of confidentiality associated to integrity (P2) is always more important than that of integrity alone (P1). It is important to underline that DTLS security impact is very similar to that of IPSec one, when security levels are the same (P2 and P3), and that the greatest impact is obtained when both IPsec and DTLS securities are associated to the same data transmission. The impact measurements concerning the loss rate are always equal to zero. This is due to the nature of the connection between the two systems of the used platform.

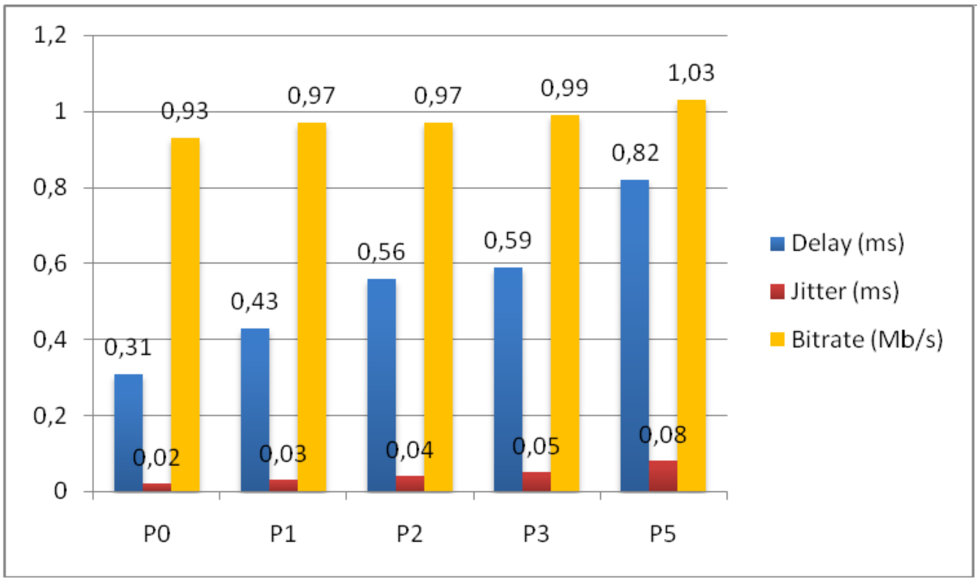


Figure 19. Security impact on QoS for UDP traffic

- *TCP traffic:* The measured parameters when securing TCP traffic are the delay and the bandwidth. Indeed, measuring loss rate for a reliable transport protocol like TCP has no meaning, because it ensures the retransmission of lost packets. As for jitter, the tools that we dispose unfortunately don't allow us to measure this parameter for TCP traffic. For TCP traffic, Figure 20 shows that the introduction of security mechanisms has almost the same effect on delay and bandwidth than that observed with UDP traffic. Indeed, delay increases when integrity is provided using IPSec (passing from P0: 0.34 to P1: 0.62). This delay reached its highest value when the TCP traffic is secured at both the "Network" and the "Transport" layers (P6: 0.93). The bandwidth also increases when introducing IPSec-AH mechanism to provide data integrity. This increase is due to the addition of the MAC in the IP packet headers. We can also note that the needed bandwidth when both TLS and IPSec (P6) are used is greater than that required when using IPSec (P2) or TLS (P4).

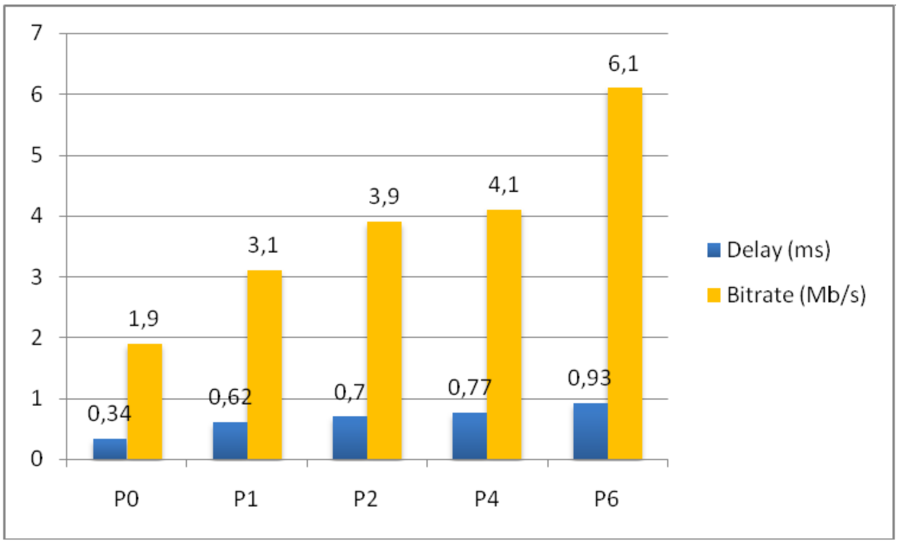


Figure 20. Security impact on QoS for TCP traffic

6. Conclusion and future works

In this paper, we have described an autonomous management architecture for NGN networks and the functioning of the SLNP negotiation protocol enabling QoS and security guaranties for mobile users communications. The SLNP protocol uses Web Services technologies that are suitable for improving interoperability in self-management environments where different wireless and wired technologies are available. The obtained results with our SLNP implementation show that interoperable Web Services usage is not at the expense of the average SLNP multilateral negotiation time. Indeed, negotiation time comparison between SLNP and COPS-SLS implementation for one round and two entities scenario [40] shows that we have obtained similar performance results with a better interoperability. The other results, concerning security impact on QoS evaluation, enables taking into account this impact while negotiating a service level including both QoS and Security.

In future works, we will try to evaluate the impact of SLNP negotiation entities number increase in a negotiation process on QoS and security negotiation time as well as the scalability of our implementation. We are using the PROMELA high level language within the Spin model checker tool for formal verification of our protocol. Finally, we are working on a second implementation for our protocol using gSOAP to validate interoperability tests.

References

[1] International Telecommunication Union – Telecommunication standardization sector, ITU-T Recommendation Y.2001, “General Overview of NGN”, December 2004.

[2] A.G. Ganek, T.A. Corbi, “The dawning of the autonomic computing era”, IBM Systems Journal, vol. 42, No 1, 2003, pp. 5-18. http://dx.doi.org/10.1147/sj.421.0005

[3] P. Horn, “Autonomic computing: IBM perspective on the state of information technology”, IBM T.J.Watson Labs, NY, AGENDA’01, Scottsdale, October 2001.

[4] R. Sterritt, D. Bustard, “Autonomic computing: a means of achieving dependability”, IEEE Engineering of Computer-Based Systems, IEEE ECBS’03, Huntsville, p. 247-251, April 2003. http://doi.ieeecomputersociety.org/10.1109/ECBS.2003.1194805

[5] R. Sterritt, D. Bustard, “Towards an autonomic computing environment”, Proceedings IEEE Database and Expert Systems Applications, IEEE DEXA 2003 workshops, Prague, p. 694-698, September 2003. http://dx.doi.org/10.1109/DEXA.2003.1232103

[6] IBM Group, “An architectural blueprint for autonomic computing”, IBM White paper, June 2005.

[7] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, “RFC 2475: An Architecture for Differentiated Services”, Request for Comments, IETF, December 1998.

[8] D. Grossman, “RFC 3260: New Terminology and Clarifications for DiffServ”, Request for Comments, IETF, April 2002.

[9] V. Sarangan, J.C. Chen, “Comparative Study of Protocols for Dynamic Service

Negotiation in the Next Generation Internet”, IEEE Communications Magazine, pp. 151-156, March 2006. http://dx.doi.org/10.1109/MCOM.2006.1607879

[10] J. De Clercq, P. De Schrijver, “PPP Diffserv SLA Negotiation”, draft IETF, Mars 2000.

[11] T. M. T. Nguyen, N. Boukhatem, “Service Level Negotiation and COPS-SLS Protocol”, Annals of Telecommunications, vol.59, No.1-2, January-February 2004.

[12] Ambient Networks Consortium, “Connecting Ambient Networks – Architecture and Protocol Design (Release 1)”, Deliverable D.3.2, March 2005.

[13] TEQUILA PROJECT, “Final Architecture, Protocol and Algorithm Specification”, Tequila deliverable D 3.4 - Part B, October 2003

[14] J. C. Chen, A. McAuley, S. Sarangan, and al., “Dynamic Service Negotiation Protocol”, draft-ietf-dsnp-01, December 2002.

[15] N. Mbarek, F. Krief, M.A. Chalouf, “Enabling Service Level Negotiation in a Self-Management Framework”, IEEE International Conference on Signal Processing and Communication, IEEE ICSPC’07, Dubai, UAE, November 2007.

[16] C. Kiss, “Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 2.0”, W3C Working Draft, W3C, April 2007.

[17] ISO/IEC TR21000-1:2004, “Information technology - Multimedia Framework (MPEG-21) - Part 1: Vision, Technologies and Strategy”, 2004.

[18] ISO/IEC TR21000-7:2007, “Information technology - Multimedia Framework (MPEG-21) - Part 7: Digital Item Adaptation”, 2007.

[19] Z. Jrad, and al., “A user assistant for QoS negotiation in a dynamic environment using agent technology”, International Conference on Wireless and Optical Communications Networks, WOCN’05, UAE, March 2005.

[20] B. Tebbani, I. Aid, G. Pujolle, “SLA-Based Dynamic Resource Management in Wireless Environment”, ACS/IEEE International Conference on Computer Systems and Applications, AICCSA’08, Networking and Multimedia Track, Qatar, April 2008.

[21] J. C. Royer, R. Willrich, M. Diaz, “User Profile-Based Authorization Policies for Network QoS Services”, Proceedings of the Seventh IEEE International Symposium on Network Computing and Applications, pp.68-75, July 2008. http://dx.doi.org/10.1109/NCA.2008.39

[22] W. Vambenepe, “Web Services Distributed Management: Management using Web Services (MUWS 1.0) Part 1”, OASIS Standard, August 2006.

[23] I. Sedukhin, “Web Services Distributed Management: Management of Web Services (MOWS) 1.0”, OASIS Standard, August 2006.

[24] M. A. Chalouf, F. Krief, “A Secured Service Level Negotiation In Ubiquitous Environments”, International Journal of Communication Networks and Information Security (IJCNIS) Vol. 1, No. 2, August 2009.

[25] S. Kent, K. Seo, “RFC 4301: Security Architecture for Internet Protocol”, Request for Comments, IETF, December 2005.

[26] T. Dierks, E. Rescola, “RFC 4346: The Transport Layer Security Protocol Version 1.1”, Request for Comments, IETF, April 2006.

[27] E. Rescola, N. Modadugu, “RFC 4347: Datagram Transport Layer Security”, Request for Comments, IETF, April 2006.

[28] A. Nadalin, C. Kaler, R. Monzillo, P. Hallam-Baker, “Web Services Security Specification 1.1”, Spécification OASIS, Février 2006.
[29] IEEE P802.21/D10.0, “Draft Standard for Local and Metropolitan Area Networks: Media Independent Handover Services”, IEEE - LAN MAN Standards Committee, April 2008.
[30] C. Ivrine, and al., “Security as a Dimension of Quality of Security Service in Active Service Environments”, Proc. Active Middleware Services Workshop, San Francisco, CA, pp 87-93, August 2001.
[31] Apache Tomcat, available at: http://tomcat.apache.org
[32] Apache Axis, available at: http://ws.apache.org/axis
[33] Sun Microsystems, “MySQL: La base de données open source la plus populaire au monde”, 2011. Available at: http://mysql.fr/
[34] Sun Microsystems, “Java: The source for Java developers”, 2011. Available at: http://java.sun.com
[35] The Openswan project, available at: http://openswan.org/
[36] The OpenSSL project, available at: http://www.openssl.org/
[37] The Sjitter tool, available at: http://www.nicolargo.com/dev/sjitter
[38] The Iperf tutorial, available at: http://openmaniak.com/fr/iperf.php
[39] The Hping tool, available at: http://www.hping.org/
[40] N. Thi Mai Trang, N. Boukhatem,Y. Ghamri, G. Pujolle, “Service Level Negotiation and COPS-SLS Protocol”, IEEE Communication Magazine, May 2002, pp. 158-165.

Copyright Disclaimer