

A Web Services Based Resource Signaling Scheme in Multimedia Satellite Systems

M. Gineste¹, B. Jacquemin¹, P. Berthou^{1,2}, A. El Fatni¹, T. Gayraud^{1,2}, C. Baudoin³

¹LAAS-CNRS ; Université de Toulouse ; 7, avenue du Colonel Roche, F-31077 Toulouse, France

²Université de Toulouse ; UPS

³Thales Alenia Space, Toulouse, France

Abstract— This paper introduces a Web Services based approach that extends resource reservation mechanism for interactive multimedia applications over multimedia satellite systems. It proposes a flexible and extensible scheme for the discovery of services in the network and the configuration of the satellite terminal when resources are requested. It finally presents experimental results of the proposed scheme over a realistic DVB-S2/RCS emulation platform.

Index Terms— Web Services, Resource Reservation, SIP

I. INTRODUCTION

Current Internet is built upon heterogeneous networks, particularly regarding access networks. Among them, satellite systems have an important role to play taking into account their broadband coverage and easy deployment capacities. However, characteristics associated to satellite systems, such as large propagation delay and asymmetric properties, require particular treatments in terms of resource management and optimization. Thus, on-demand Quality of Service support over satellite communications is of primary importance for multimedia applications, in particular on the return link. Indeed, a large number of distributed satellite terminals (providing Internet access to end-users) might share a limited and on-demand uplink resource and thus the return link might become a bottleneck.

Numbers of research and standardization tracks have developed solutions to allow dynamic resource allocation in multimedia satellite systems (ETSI BSM, Satlabs,...). The principles of a centralized resource allocation mechanism (DAMA), multiple layer 2 services based on different capacity request types, differentiated services for the IP layer and mapping of layer 3 services over layer 2 are broadly admitted and partially standardized [9][10].

However, the mapping of applications QoS requirements toward available services offered by the network is still an open issue. Some are considering a static configuration done by the network administrator, some are considering a dynamic

approach or an automatic approach based on Internet session signaling protocol (SIP) [20].

[1][2]. Anyway, one of the remaining problem consists in determining the qualitative QoS parameters (such as bandwidth and temporal constraints) to reserve for a given session.

This paper presents a web services based solution to this problem. It offers an easily deployable solution thanks to open standards and a good opportunity to introduce new codecs or services in the network thanks to the dynamicity of web services. In the opposite of what could be expected, it offers quite acceptable performances over satellite links.

This paper is structured as follows. The following section details resource reservation mechanisms in satellite systems. Section III presents web services principles. Then, in section IV our new services for resource reservation are explained. Finally, section V presents a performance evaluation of this architecture over a satellite emulation platform.

II. RESOURCE ALLOCATION IN MULTIMEDIA SATELLITE SYSTEMS

A. DVB-S2/RCS Network Architecture

The scenario, shown in Figure 1, gives an overview of the satellite network architecture in a mesh topology. The main components of the network are a geostationary satellite with onboard switching capabilities, Satellite Terminals (ST) which provide single PC or LANs with the access to the network, and a Gateway (GW) which allows the connection with Internet core networks. In this configuration, the GW plays the same role than the STs; it is thus represented by a ST allowing access to Internet. The uplink access from each ST is managed through DVB-RCS interfaces. STs and GW are boundary devices between the satellite and terrestrial links and play an important role in access to satellite resources and hence in QoS provisioning. Both devices implement IP routing and have an IP interface on the satellite segment, as IP serves as a common denominator between the satellite and terrestrial networks.

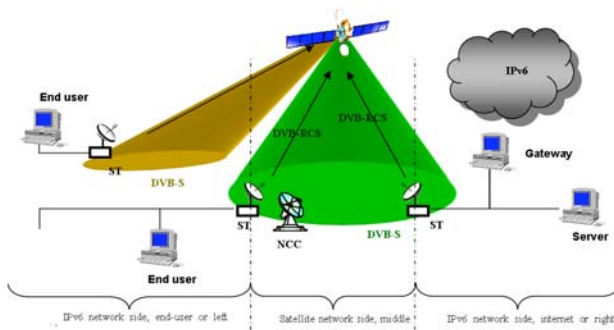


Figure 1 - Satellite Network Architecture

The return link access scheme in DVB-S/RCS systems is MF-TDMA (Multi-Frequency Time Division Multiple Access). The return link is segmented into portions of time and frequency ("superframes"), each of which is divided into timeslots ("bursts") of either fixed or variable durations during which STs are able to transmit MPEG2-TS packets or ATM cells.

The entire satellite system control, especially STs' synchronization and resource allocation, is performed by the Network Control Center (NCC). It periodically broadcasts a signaling frame, the TBTP (Terminal Burst Time Plan), which updates the timeslot allocation within a superframe between every competing ST. This allocation can be dynamically modified depending on Satellite Terminals' requests thanks to a bandwidth on demand protocol called Demand Assignment Multiple Access (DAMA). It supplements the STs with the ability to frequently request capacities that fit their current respective traffic load to the NCC.

B. Resource Reservation principles

1) User tunable

Very few applications currently implemented are aware of the QoS provided by the underlying network. As the applications are rarely able to define their own requirements, a user-oriented solution, called « QoS Agent », has been defined in the SatIP6 project [6]. Implemented on the user terminal, the QoS Agent detects the outgoing traffic flows and thanks to a graphical interface listing them, the user is able to assign one of the available QoS services and, through interactions with the QoS Server, to remotely configure the multi-field classifier in the RCST, for only one or for all the streams the application uses. In fact, each time a service is selected by the user for a given application, the QoS Agent sends to the QoS Server the concerned connections' list with the reference of the associated service using a dedicated transactional protocol. Using the received information, the QoS Server is able to tag and redirect these packets coming from the user terminal towards the appropriate requested IP service. If resources are not available, the QoS Agent is immediately informed. This service is available for clients who have specific contracts with the service provider of the satellite network.

2) QoS-aware SIP proxy

An alternative solution to the QoS Agent is to automate the resources reservation and make it transparent to users who wouldn't be able to choose the most appropriate classes of service to the different flows (audio and video for instance). In order to do this, the Session Initiation Protocol (SIP) has been chosen because of its growing success in both public (open source code) and proprietary domains. This mechanism allows configuring the QoS transparently by the analysis of SIP messages being exchanged by applications. This analysis is made by an enhanced SIP Proxy (or QoS-aware SIP Proxy).

The enhanced SIP Proxy intercepts the session descriptors included in SIP messages, obtains the characteristics of each media involved in the session and pass it to a QoS Agent communicator in charge of (automatically) reserving and releasing QoS for the session via the QoS Server (more detail is provided in sections IV.A.1) and V.B).

Besides traditional SIP communication, additional functionalities are thus required:

- An SDP analyser making the Proxy able to analyse the session descriptions
- A table of medias updated during the session establishment. The medias negotiated between the caller and the callee are identified by a call-ID.
- An SDP/Diffserv mapping
- A QoS module (the QoS Agent communicator) which realizes the resources reservation associated to each media using the QoS Server.

In the proposed architecture, a QoS-Aware SIP proxy is deployed in each user LAN. This distributed architecture is well suited for an access or mesh topology based on a regenerative satellite because it addresses the following two concerns:

- Scalability concerning flow QoS management in user LAN;
- Session establishment delays: the number of round trips of session and QoS signalling on the satellite link are minimized.

C. Quantification and Discovering Problem

Two main issues can be raised in QoS-Oriented architectures such as the one described in the previous sections: 1) The precise quantification of the QoS to reserve in accordance to the session being initiated (quantification of application requirements), 2) the discovering of available services in the architecture and their configuration, depending on applicative requirements.

The first issue is related to the quantifications of the QoS to reserve. Although the applicative QoS requirements has been widely studied in the literature and in standards [11][12][13], the current information on QoS, provided by applications is poor. Precise information are required for the network to configure mechanisms and guaranty the appropriate QoS to the applications: these information include bandwidth requirement, tolerated delay, jitter, and packet losses, dependency of sessions' flows, possible priority of flows, possible adaptation of the session (specification of min and max characteristics for an acceptable session quality). However, most of current applications advertise only technical information to initiate the session, via session control protocols (e.g. SIP [1], SDP [14], RTSP [15]), such as

addresses and ports, codec type and optionally bandwidth. We will particularly focus on this first issue in the paper.

The second issue is typical of auto-adaptive network in a mobility environment. Agreements can be passed among various access providers but they don't necessarily implement their QoS-oriented services the same way. If we consider a mobile user moving from an access network to another one (e.g. changing between WLAN access points serviced by two different STs) or a new entering user in the network, continuity in the QoS delivered might be agreed. The user (via a QoS Control plane) might want to advertise the information on the required QoS of the current sessions and appropriate configuration of services in the new access network has to be performed.

To address these issues generically, we propose to use Web Services concepts; thus, this solution can be applied to other types of architecture. Our aim is to provide a mapping from high level session description toward precise QoS parameters and to discover and bind easily to the QoS-oriented services offered in the satellite system. A brief presentation of the Web Services paradigm is presented in the next section. Section IV presents the contributions and section V evaluates different scenarios of the proposal.

III. WEB SERVICES

A. Overview

Web Services are encapsulated, loosely coupled contracted functions offered via standard protocols.¹ Web Services are at the confluence between traditional software and on-line applications development methodologies: they inherit dynamic and adaptable characteristics from the Internet and capitalize on paradigms implemented in norms like CORBA. Web Services extend these paradigms by transforming Internet into 1) a hyper repository for software components interfaces and 2) the very location where takes place all inter-process binding.

B. Architecture and Principles

Web Services architecture requires three fundamentals operations: publish, find and bind; and three actors each playing a specific role in the Web Services loop: 1) The Service Provider offers a service and publish it to the broker. 2) The Service Broker hosts all the necessary information to advertise and bind to the services located on Providers. 3) The Service Requester finds a required service using a Broker and then bind to it directly on the Provider.

Figure 2 shows the three operations performed by the three actors of a Web Service:

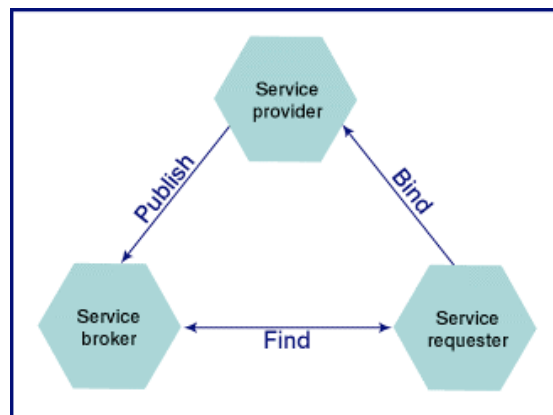


Figure 2 - Web Services Paradigm

C. Standards and protocols

XML-based standards were developed to normalize the three operations of the Web Services architecture. **SOAP** allows sending and receiving messages over the Internet without making any reference to any specifics of the application environment or middleware. In order to do that, SOAP is encoding inter-machine function calls and response in XML so that they can pass among heterogeneous systems (independently of the operating system and of the programming language used). It creates a generic interface to access remotely any local applications. This protocol is used in the binding part between Service Requesters and Providers. **WSDL** (Web Services Description Language) is an XML schema used for describing the functions exposed by web services. It is located on the Service broker and used in the finding operation. **UDDI** (Universal Description, Discovery, Integration) defines a Registry where Web service providers can register services and where developers and applications (Service Requesters) can look up services. UDDI represents the Service Broker in the architecture. UDDI registries are periodically synchronized to provide an up-to-date image of available services.

IV. WEB SERVICES SATELLITE SYSTEMS

A. New Services for Resource Allocation

This section presents a solution for the quantification, discovering and configuration problems given previously. It uses a Service Oriented Approach based on Web Services. This paradigm guarantees the better flexibility and extensibility for new protocols, but on the other hand of more overhead and possibility problems of performances. The section V analyses the performances over a satellite link and shows that the presented solution is not unrealistic.

In fact, two new web services are specified to improve the resource allocation mechanism of Multimedia Satellite Systems. A detailed description of these web services is provided in the following sections.

1) Media Type Repository

The first Web Service is proposed to address the quantification issue of QoS requirements of the applications. A repository

¹ Definition of Web Services from WebServices.org

containing media type specifications will be used to resolve this quantification issue.

Media type specifications have been defined in [16] and are designed to provide a library of standard media descriptions. These specifications may include two types of information: generic information of application types (e.g. tolerable end-to-end delay or internal structure of a flow) and specific QoS information of particular applications (e.g. encoding bitrate). Two sources of information might be considered to feed the media type specifications:

- First, concerning generic information by types of application, quantitative evaluation of QoS requirements has been defined in [17]. Parameters specified in this recommendation are temporal constraints (tolerable delay and jitter) and possible packet loss rates.
- Secondly, concerning specific application requirements, standards or application developers are able to provide parameters such as: peak bit rate and burst, ADU (Application Data Unit) size. Redefinition by developers of temporal constraints and packet loss rate tolerance specific to a particular application or codec is also feasible.

An example of media type specification for a GSM audio flow is depicted on Figure 3.

```
<?xml version="1.0" ?>
- <definitions xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="http://dmi.ensica.fr/xqos/xqosMediaType.xsd">
- <mediaType id=" GSM ">
  <param name="peakBitRate" value=" 13.2kbps" />
  <param name="sustainableBitRate" value=" 13.2kbps" />
  <param name="maximumPacketSize" value=" 73bytes" />
  <param name="maximumBurstSize" value=" 73bytes" />
  <param name="targetDelay" value=" 150ms" />
  <param name="maxDelay" value=" 400ms" />
  <param name="maxJitter" value=" 1ms" />
  <param name="maxLoss" value=" 5%" />
</mediaType>
</definitions>
```

Figure 3 – Media Type specification for GSM audio flow

A permanent access to the media type repository is required for sessions being initiated as well as for administrator and application developer to expand the media type base. The web service proposed offers then two entries: a management

function and a consultation function. From the management point of view, new media types might be added by a restricted number of network administrators and new applications' developers. An *addMediaType method* and a *deleteMediaType method* are provided for this purpose. From the consultation point of view, parameters specific to an application might be requested by any session being initiated. A *getMediaDescription method* is thus provided.

The integration of the Media Type Repository in the architecture and in the QoS reservation process of sessions' initialisation is depicted in Figure 4. The QoS-aware SIP proxy (cf. section II.B.2) is in charge of reserving QoS for the sessions and thus is responsible of sending request to the Media-Type repository when sessions start. When receiving the SDP description of the session, within SIP session progress message (cf. section V.B), the codec name, as specified in SDP, is retrieved and a SOAP request to the *getMediaDescription* method including this argument is sent to the MTR. During session establishment, SDP provides audio and video codec numbers and name that are standardized by the IANA organism [18]. MTR performs a mapping between these numbers and the codec comprehensive name. Finally, the full description of the media, including quantified QoS requirement for the flow, is sent in response to this request to the SIP Proxy. With this information in hand, the SIP proxy is able to start the QoS reservation for the corresponding session in the Satellite System (cf. next section).

The MTR must be available at any moment to provide media specification when sessions are being initiated. Local copies of the repository might be implemented and updated on a regular basis, in order to improve the access performance to the repository. The initiation of session and QoS reservation is not highly time critical (because it is done before the session starts); however a too large initialisation time might affect the user experience, particularly in the satellite context where round trip delays are high. Evaluation of performance using a centralised solution and local copies of the repository is realized in section V.C.

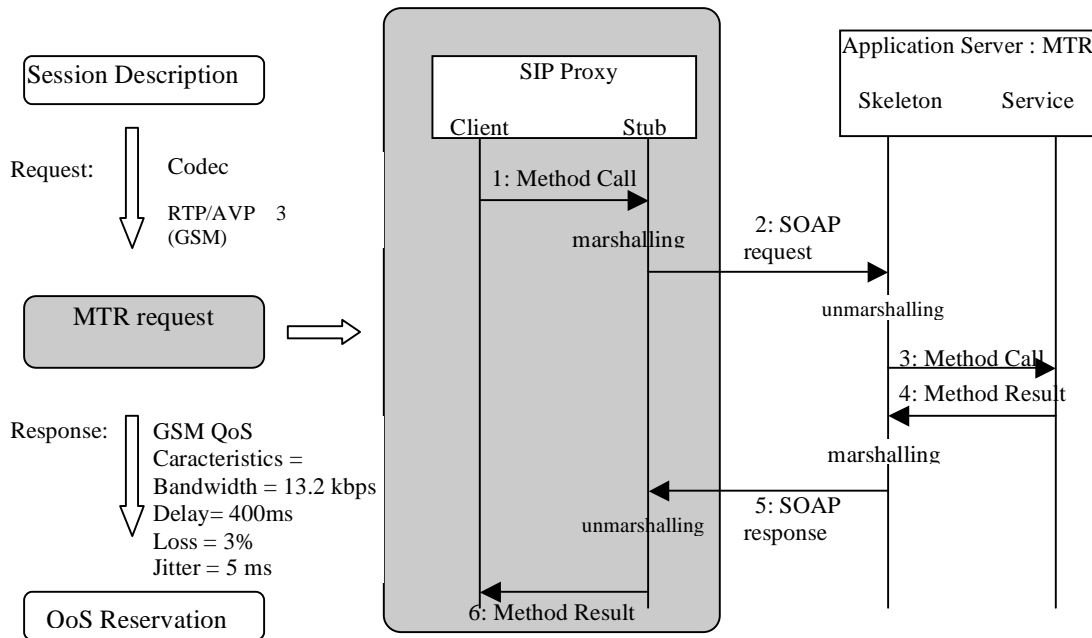


Figure 4 - MTR Web Service

2) Satellite Terminal Configuration

This Web Service was initially developed during the ESA funded TRANSAT project [3]. It describes an easy and flexible scheme for satellite terminal configuration. This service proposes a description of the current status of QoS availability on the Satellite Terminal and a dynamic reconfiguration of the DiffServ node for admitted flows. A WSDL description of the binding information to the provided service and the advertised methods, is stored on a centralized UDDI registry (replications of these registries might be deployed). This centralized registry enables discovering the service easily, independently of the implemented QoS architecture. Any QoS management and control plane could discover on-the-fly and bind directly to the service. This property of the web service is useful in mobile and unknown environments or to control the QoS configuration from a remote location. However, a level of security in the service discovery and usage must be added to insure authentication process of the users of the service. This is out of the scope of this paper.

Three methods are provided by the Service: *requestQoSAvailability*, *reserveFlow*, *releaseFlow*.

Here is described a possible usage of the Web Service, in the context of the previously presented QoS architecture:

1) When a new session is being initiated (signaled by the SIP Invite method), the QoS Agent communicator located on the QoS-aware SIP proxy checks availability of the resource using the *requestQoSAvailability* method.

2) After gathering the quantified QoS requirements of the session using the MTR service, the QoS Agent communicator verifies if the new flows of the session can be admitted. If so, it reserves the required QoS for each flow of the session, using the *reserveFlow* method of the web service.

3) The service on the Satellite Terminal reconfigures the Diffserv node accordingly.

4) When session ends (signaled by the SIP BYE message), the QoS Agent sends a *releaseFlow* method request for each flow of the session. The ST releases then the resource accordingly.

V. PERFORMANCE EVALUATION

In this part, we will first present the satellite emulation platform and then the experimentations that have been tested on it.

A. PLATINE: the satellite emulation testbed

The Satellite network emulation testbed (PLATINE) has been initially developed in the SATIP6 IST project. It is compliant with the architecture adopted within the ETSI BSM [5] group and the DVB-RCS and DVB-S2 standards. It is able to emulate complex scenarios for next generation satellite network. Each network element involved in the satellite network is emulated in the platform on a dedicated node. In fact, 3 users' LANs of one or two nodes (standard Linux/Windows systems) are connected to the emulated satellite network with 3 ST (standard Linux system) that implement an almost complete DVB-S2/RCS stack. The satellite core network is emulated thanks to the Satellite Emulator (SE) as link emulator and the Network Control Center (NCC) for bandwidth management (DAMA). 9 computers are used as described in Figure 5.

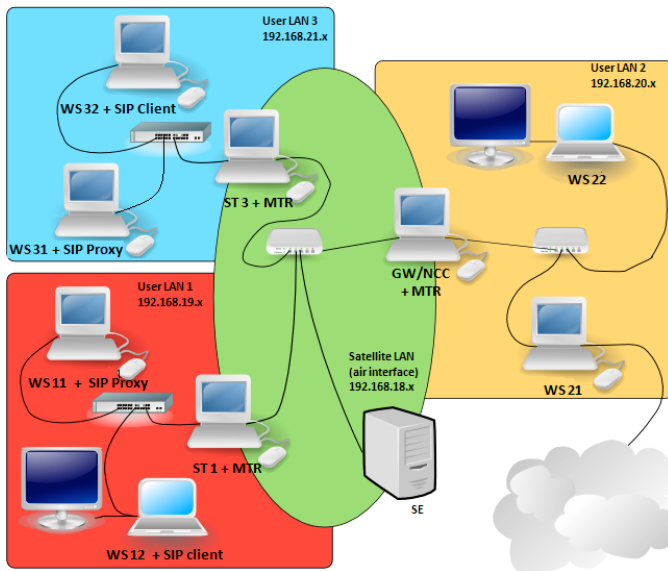


Figure 5 - Satellite emulation testbed

In order to have the most modular platform and so preserves room for future evolution (GSE), stringent requirements were fixed before the development phase.

The emulation testbed takes advantage of a linux system (Fedora Core 5) which natively supports IPv6 and a wide panel of IPv6 applications (*Apache* as HTTP Server, *Mozilla* as HTTP Client, *Vsftpd* as FTP Server, *Gnomemeeting* for Videoconferencing, *VideoLanClient* for Videostreaming), as well as advanced network and QoS features.

The different building blocs of the testbed are the following:

- The *satellite carrier* package is responsible for the different satellite carriers emulation on top of Ethernet (DVB-RCS, DVB-S2 and Signaling Channels) and the simulation of typical satellite bit errors and delay
- The *DVB-S2/RCS* package implements a framing structure compliant with the DVB-S2/RCS standards and fills DVB-RCS frames with upper layer packets (ATM or MPEG2-TS) coming from the ENCAP bloc layer. In order to achieve proper QoS, this layer manages synchronization and queues according to the authorizations a DAMA algorithm delivers.
- The *DAMA* package implements the DAMA algorithms used to manage the satellite resources allocation at layer 2 taking into account adaptive physical layer information.
- The *ENCAP* package implements AAL5 and ULE encapsulation schemes, and is in charge of the segmentation and reassembly functionalities (ATM or MPEG2-TS).
- The *IP QoS Package* implements common mechanisms to enable differentiation at this level. It mostly relies on QoS services offered by Linux kernel, retrieves incoming packets from IP network with their associated tag and forwards them to the lower layer.

As described in the Figure 5, in order to make our experimentations, two SIP clients have been installed in the work stations WS 32 and WS 12 and two SIP Proxies in WS 31 and WS 11. Moreover, one MTR has been deployed on each ST/GW.

B. Establishment of a SIP session with MTR

The Figure 6 shows how the MTR mechanisms have been implemented in the process of establishing a SIP session compliant with RFC 3312 [2] which specifies the integration of resource management in a SIP session. In this case, a distributed scenario is presented: an MTR is placed on each user LAN along with the ST. First, the caller (WS12) sends an INVITE message with an SDP including a set of codecs (sdp1). On the callee side (WS 12), the user agent answers with a Session Progress message which specifies a new SDP (sdp2) with the acceptable codecs for the session (i.e. the codecs available on this client). When the Proxy located on WS 31 receives the message, a SOAP request is sent to the MTR to know the characteristics of the codecs (peak bitrate, maximum loss, maximum delay and maximum jitter). After the SOAP response, the QoS Agent Communicator on the SIP proxy side chooses the most appropriate codec for the session (depending on the service provider policy) and initiates the reservation process. The Session Progress message is forwarded to the next proxy with a new SDP (sdp3) which contains the selected codec on ST3 side and the result of the QoS reservation. The same process is realized on the ST1 side and QoS is reserved (if possible) for the corresponding codec. The Session Progress is transmitted to the SIP client with the result of the QoS reservation process on ST1 side. Then, the client sends an UPDATE message to give to the callee the selected codec and the results of the reservation.

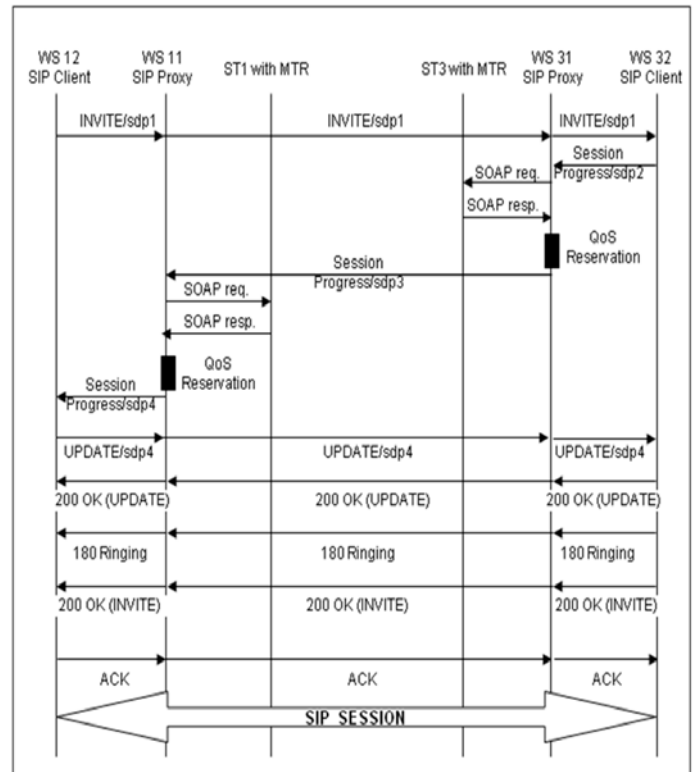


Figure 6 - Establishment of a SIP session with MTR in distributed scenario

C. Evaluation of the response time

In this part, two scenarios will be evaluated on our satellite emulation platform. In each case, we consider a communication between two SIP clients located behind two different ST as shown on Figure 6. The first scenario corresponds to a centralized scenario where the MTR is located in the gateway of the satellite system and the second one corresponds to a distributed scenario (the one presented on the Figure 6) where an MTR instance is located in each ST/GW. Before and after the SOAP request/response, the messages to establish and to close the TCP session are not shown. A set of measures of the response time (25 repetitions) have been made for each scenario and the following table presents them.

	Mean time of response	Minimum time of response	Maximum time of response
Scenario 1	1270 ms	1229 ms	1335 ms
Scenario 2	17.75 ms	12 ms	29 ms

Table 1. Response time in centralized and distributed scenarios

We first remind that the one-way delay to cross the satellite is about 250 ms. In the first scenario, the mean time of response corresponds to the three-way handshake of TCP (about $3 \times 250\text{ms} = 750\text{ms}$) + the request/response of SOAP (about $2 \times 250\text{ms} = 500\text{ms}$) to obtain the codec characteristics. Although this solution is simpler to implement, it is penalizing to add more than 1s to the session initiation time, particularly if the session has to be re-initiated to change the codecs or in a mobile context where the session has to be resumed in the new location [4]. In the distributed scenario, only about 17.75 ms are necessary to obtain the characteristics for each proxy so it takes 35.50 ms, which is insignificant in comparison with the session initiation time particularly in a satellite context. Indeed, if we consider the interval between the time where the callee is picking up the SIP phone and the time where the communication is beginning, it takes at least 500 ms (OK + ACK). The conclusion of this experimental evaluation is that the distributed scenario offers much better results than a centralized one in the satellite context. Thus, the offline synchronization of these local copies of the MTR enables to keep an up-to-date media types base and to reduce substantially the session initiation duration as well as renegotiation in a mobility context.

VI. CONCLUSION

Web Services give a good opportunity to deploy easily new services in satellite systems. This article shows how it can be used to help the resource reservation process to adapt to new codecs. Another example is given with the satellite terminal configuration. It shows how a satellite operator can use it to allow easy changes of satellite terminal configuration. Moreover, web services provide a completely opened and standardized API that will facilitate the integration of satellite networks in global NGN infrastructures. The example of Parlay X [7] that is a set of telecommunications Web services and that is part of the IP Multimedia System [8] confirms this evolution. Experimental evaluations show that even when

using TCP, web services are still effective over long delay networks in a distributed context.

REFERENCES

- [1] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, "SIP: Session Initiation Protocol," June 2002, IETF RFC 3261
- [2] G. Camarillo, W. Marshall, J. Rosenberg, "Integration of Resource Management and Session Initiation Protocol (SIP)", IETF RFC 3312, October 2002
- [3] M. Gineste, P. Sénac, "Quality of Service Solutions in Satellite Communication", 4th IEEE International Conference on Networking, April 2005.
- [4] H. Schulzrinne and E. Wedland. "Application-layer mobility using SIP", ACM SIGMOBILE, Vol. 4, Number 3, July 2000, pp.47-57.
- [5] TR102157 v1.1.1. ETSI TC SES; Broadband Satellite Multimedia; IP Interworking over satellite; Performance, Availability and Quality of Service, July 2003
- [6] IST SATIP6 Project (Contract IST-2001-34344), <http://satip6.tilab.com>
- [7] Parlay X Web Services Specification, Version 3.0, June 2007, <http://www.parlay.org/en/specifications/pxws.asp>
- [8] G. Camarillo, M.A. Garcia-Martin. The 3G IP Multimedia Subsystem. West Sussex: Wiley, 2004
- [9] DVB-RCS: Digital Video Broadcasting (DVB); Interaction channel for satellite distribution systems, ETSI TR 101 790 V1.3.1 September 2006.
- [10] ETSI Technical Specification. Satellite Earth Stations and Systems (SES); Broadband Satellite Multimedia (BSM): QoS Functional Architecture ETSI TS 102 462 V1.1.1 December 2006.
- [11] ISO/ITU-T Recommendation X.641 Information Technology, Quality of Service Framework [ITU-T Recommendation X.641 | ISO/IEC 13236], 1997.
- [12] ISO, "Quality of Service Framework", ISO/IEC JTC1/SC21/WG1 N9680, International Standards Organization, UK, 1995.
- [13] Andrew T. Campbell. "A Quality of Service Architecture. PhD thesis, Computing Department", Lancaster University, Jan. 1996.
- [14] M. Handley and V. Jacobson, "RFC 2327: SDP: Session description protocol," Apr. 1998.
- [15] Schulzrinne H., Rao A., and Lanphier R. "Real Time Streaming Protocol (RTSP)". IETF RFC 2326, Apr. 1998.
- [16] E. Exposito, M. Gineste, R. Peyrichou, P. Sénac, M. Diaz, "XQOS: XML-based QoS Specification Language", The 9th International Multi-Media Modelling Conference, Taiwan, January 2003.
- [17] ITU-T Recommendation G.1010 "Catégories de qualité de service multimédia pour l'utilisateur final", Novembre 2001.
- [18] Internet Assigned Numbers Authority, unit of Internet Corporation for Assigned Names and Numbers (ICANN).
- [19] S. Combes, O. Alphand, P. Berthou, T. Gayraud, "Satellite and next generation networks: QoS issues", Journal of space communications 2005, Volume 20, Number 3-4 (2005).