SIP-based QoS Control over Satellite Networks

Rehan Inam Qureshi
Department of Computer Engineering
Sir Syed University of Engineering & Technology, Karachi, Pakistan

Olivier Alphand
Software & Tools for Communication Group
CNRS-LAAS
Toulouse, France

Abstract—Satellite access networks (SAN) are expected to play a decisive role in bridging the digital divide through providing broadband access to multimedia services in geographical areas where terrestrial access networks like cable or ADSL cannot be deployed. SANs currently lack a clear QoS architecture in order to fit seamlessly into the end-to-end NGN framework. Several architectures mixing IP QoS layer mechanisms with specific satellite MAC dynamic allocation schemes are already designed. However, very few multimedia applications can actually benefit from these architectures since most of them are not QoS aware. This work presents a SIP based multimedia signaling architecture where existing SIP applications can benefit from a QoS enabled IP network without undergoing any adjustments or modifications. In addition to that, the realization of the proposed solution is also described along with its performance evaluation tests and their results on an emulated satellite platform, providing DiffServ services, which was implemented during the European IST SATIP6 project.

Index Terms—Multimedia Communication, SIP, Quality of Service, Satellite

I. INTRODUCTION

Considered as one of the key access technologies that is able to reduce the digital divide, geostationary access satellite networks will be integral part of the future global NGN architecture. These satellites were traditionally used only for broadcasting but now with the standardization of DVB-RCS [1], bidirectional services can also be provided. This allows connecting the users to core networks in remote areas where the terrestrial infrastructure either does not exist or is difficult to manage.

The satellite networks while benefiting from being more accessible also have some disadvantages. They have higher costs, longer propagation delay, more frequent transmission errors and lesser resources as compared to the terrestrial networks. Therefore in order to satisfy the user, especially in context of multimedia communication; the delays, the bandwidth and the error rates must be controlled which involves adequate management of quality of service (QoS).

The quality of multimedia communication through satellite networks can be assured via advance QoS management capabilities such as QoS aware network equipment, admission control techniques and interaction between call signaling, resource management and admission control. Unfortunately, for satellite networks, no architecture has been standardized that uses such QoS management features to facilitate the use of DVB-RCS. The QoS applications available today are largely not network aware and consequently do not address the specific problems of satellite networks. Eventually this leads to system specific or proprietary solutions.

This paper presents a QoS solution for satellite networks using Session Initiation Protocol (SIP) [4]. The emphasis is made on reducing the signaling load from the satellite network while providing a generic architecture compliant with increasing legacy SIP applications. The following sections also give an account of the implementation details and evaluation results.

II. ARCHITECTURE OVERVIEW

The reference scenario of the work being presented here, depicted in the Figure 1, is based on a satellite network implementing a DiffServ IP QoS framework. The SIP Proxy and Clients are connected through overprovisionned access networks to the satellite network. The network Edge Routers (ERs) are in effect Satellite Terminals (ST) which provide access to the satellite network. One access network could be interconnected with the Internet backbone and thus would host a special ST called Gateway (GW). The QoS provided in the core network is accessed via QoS Server that is assumed to coincide with ERs. The testbed implementing this scenario comes from the SATIP6 project that emulates a full regenerative DVB-RCS/S satellite emulator including a stringent QoS framework. For further details please refer to [2]-[3].

Figure 1: Reference Scenario

When DiffServ is being implemented in the core satellite network, IntServ approach can be applied in the access networks where a dedicated protocol is used to specify the desired resources. Although this architecture represents a significantly scalable solution, it requires client software to be aware of both call signaling mechanisms and QoS signaling mechanisms. SIP functions independently from QoS mechanisms; hence legacy SIP clients are incapable of signaling QoS requirements. Thus, it is required that SIP clients should not be involved with the QoS related functions, and that these functionalities should rather be moved to some other point in the network.

As the reservation of network resources requires information like the IP address, port, and session parameters
of the callee and caller, a SIP proxy can prove to be the best alternative to a SIP client for making QoS reservation since all session signaling messages pass through the proxy. However this requires a proxy to be able to read and understand not only the SIP messages but also the SDP messages carried by SIP.

A. QoS Considerations for the Architecture

For providing a certain level of QoS, resource management can be performed according one of the following three modes:

**Best effort:** This classic mode used by internet today which is not much suited for internet telephony.

**QoS-Enabled:** A "QoS-Enabled" session allows the endpoints to complete the session establishment either with or without the desired resources. Such session will use dedicated resources if available, and use a best-effort connection as an alternative if resources cannot be dedicated.

**QoS-Assured:** A "QoS-Assured" session will establish only if all required resources are available and assigned to the session. A provider may choose to block a call when adequate resources for the call are not available.

For an Internet Telephony service to be successfully used by a large number of subscribers, it must offer a certain level of connection quality. Among others, it generally requires that the system must minimize the post-pickup delay. The “QoS-Assured” model provides tight QoS control at the cost of higher session setup delay. This may reduce the call defects but results in greater post-pickup delay. On the other hand the QoS-Enabled model provides QoS for the session without delays to the setup reducing the post-pickup delay as compared to QoS-Assured. However this may result in the initial portion of the session being impacted by impairments from not having QoS, such as voice clipping and distortion. All in all the QoS-Enabled provides a good compromise between the guarantee less Best effort mode and the QoS assured mode with very strict QoS requirements without much effecting the call setup time and providing better quality to the multimedia sessions whenever possible. Keeping this in mind, “QoS Enabled” mode is considered for the work being discussed here.

Another requirement for better performing Internet Telephony service is that the system must minimize the post-dial delay. In case of SIP, minimizing the post-dial delay requires minimizing the message processing time of SIP proxy. When QoS mechanisms are implemented with a SIP proxy, it affects the proxy performance as the proxy has to read the SIP/SDP messages and convey the gathered information to the QoS provider. This performance degradation can be kept at minimum by implementing different software optimization techniques.

B. Architecture Design

To make a SIP proxy understand an SDP message, gather information from it and make it able to use this information, it requires extensions or enhancements of the proxy. Once a proxy is able to read the SDP messages, the problem is not solved completely. The SIP clients negotiate the media parameters before the establishment of actual media session. This negotiation involves sharing of capabilities by the participating clients and their final acceptance or rejection regarding a type of media offered to be used. This offer-answer model makes the job of proxy more complicated as no single SDP packet can be examined to have the complete information for making resource reservations. Instead SDP messages should be observed in context with preceding or following messages so that complete session information can be gathered which increases the processing time and memory requirements.

For a SIP proxy to intercept all SIP messages exchanged between caller and callee, the SIP proxy should be “Stateful”. The stateful proxies keep a record of active SIP session while the stateless proxies treat each message without relating it to the previous messages of the same session. Moreover, to make the architecture well adapted to the satellite network, UDP is chosen instead of TCP to carry SIP messages. UDP has proven to be more suited for the satellite link as the round-trip time of the satellite link is around 500ms that makes TCP handshake a lengthy process. However, the SIP stack of proxies still needs to be tuned since default retransmission timers are close to the satellite RTT. Without this, retransmission mechanisms could not make the difference between a signaling packet loss or a long satellite propagation delay.

![Figure 2: Call Setup and QoS Message Flows](image)

The SIP and QoS message flows are shown in Figure 2. The SIP terminals located in the access networks are standard SIP clients and explicit proxy configurations are made in them. The session setup starts with an INVITE message sent by the caller to the local QoS aware proxy. The proxy extracts the required parameters and forwards the message towards the
invited callee. The same procedure is followed when the message arrives at the proxy of the callee’s access network.

When the callee responds with a 200 OK message, it is passed back to the callee’s proxy server. Information extraction from this message is performed in the same manner as it was done for INVITE. At this point the proxy at the callee’s side has all the information to request a specific QoS reservation from the QoS server on the callee’s access network for the callee to caller traffic flow. When the caller’s proxy receives the 200 OK message, it follows the same steps as followed by the proxy on the other end. This proxy also requests for a QoS reservation from the QoS server of its network for caller to callee traffic flow.

As a QoS Enabled model is followed here, both of the SIP proxies continue with the message forwarding irrespective of the response from the QoS provider that whether it was able to provide the requested resources or not. Moreover all of the information gathered from the INVITE and OK messages is stored until the concerned session is terminated.

III. IMPLEMENTATION

The designed architecture allows the use of generic SIP Client software; therefore for testing purpose Windows Messenger and NIST-SIP Communicator [5] were used as SIP Clients. For a SIP Proxy, the NIST-SIP Proxy [5] is chosen to be changed to incorporate the requirements of the presented architecture. One of the major reasons for choosing this proxy is that it implements the SIP standards using JAIN-SIP-1.1 API rather than propriety solutions. Special care is taken to make sure that minimum changes are made in the existing code to have the enhancements. Moreover, the new modules are independent of the existing implementation of the proxy and thus can be easily integrated with any SIP proxy developed under JAIN-SIP API.

Following are the functionalities added to the standard SIP proxy to make it a QoS aware proxy.

A. Information Extraction

Before reserving resources for a multimedia session, we must be able to answer certain questions like what are the addresses of the machines participating in the concern session, what port numbers they are going to use, what kind of multimedia communication will take place, will it bidirectional or unidirectional etc. This information can be found in the set of messages that is exchanged for setting up a multimedia session.

When a SIP message carrying an SDP message arrives at the proxy it is first sent for information extraction in the Information Extractor module. Following parameters are extracted from the SIP message as well as from accompanying SDP message.

• Call-ID from SIP / Call ID Header
• Address from SDP / Session Description / “o=” line
• Media Type from SDP / Media Description / “m=” line
• Media Port from SDP / Media Description / “m=” line
• Media Formats from SDP / Media Description / “a=” line

B. Media Table

The parameters needed for QoS reservations cannot be obtained from a single SIP/SDP message but are gathered in view of preceding or following messages. To hold these parameters, a media table is used that is updated at each SIP session establishment and destruction. Each media table is defined by a 4-uple (IP source address, IP destination address, source port, destination port), the kind of media (audio, video...) and the Real-Time Protocol (RTP) profile. The media negotiated by caller and callee can be accessed by the Call-ID, which is a unique identifier of the related SIP session thus acts as the primary key.

C. QoS Reservations

The QoS reservations and release of reserved resources is done by sending RSV message and FREE message respectively to the QoS server. A RSV message and FREE message is sent for each media type the caller and callee wishes to communicate with. The formation of such messages and sending them to the QoS server is handled by the QoS Reservation module. A RSV or FREE message is sent to the QoS server using a TCP connection. A TCP socket is first opened using the QoS server’s IP address and the port number it is listening on for the requests (usually port number 4321), afterwards the message is sent.

IV. EVALUATION

A. End to End QoS Evaluation of the EF Services

In order to illustrate the temporal constraints achieved by the EF (Expedited Forwarding) service on the satellite segment, we perform the following experiment on the SATIP6 satellite emulation platform. 4 simultaneous Voice over IP (VoIP) sessions classified in the EF service were generated over the satellite network under different Satellite Terminal loading conditions. It consists in a homogeneous (considering the codec: GSM and the RTP packetization properties) audio traffic aggregation. The results are illustrated in Figure 3.

Figure 3: EF Satellite End to End Delay Evaluation

Whatever the loading conditions or the aggregation scheme, we observed that admitted flows are protected from congestion and approximately benefit from the same QoS
within the EF service: bounded delays [250 ; 320ms] and jitters [-40 ; 20ms]. These performances come out to be relevant in a satellite environment considering the VoIP requirements.

B. Round-Trip Time of SIP Messages

Roundtrip time of SIP messages was observed in an effort to identify how it is affected by different loading conditions. The roundtrip time was calculated for three message sets, INVITE-RINGING, OK-ACK and BYE-OK (see Table 1). This means for INVITE-RINGING time, the difference of time between sending an INVITE and receiving a RINGING message is observed. These three sets of messages were chosen for testing as they involved no interaction with human beings and thus no inconsistent delay.

<table>
<thead>
<tr>
<th>Loading Conditions</th>
<th>INVITE-RINGING (sec)</th>
<th>OK-ACK (sec)</th>
<th>BYE-OK (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>60%</td>
<td>0.75</td>
<td>0.71</td>
<td>0.68</td>
</tr>
<tr>
<td>80%</td>
<td>0.75</td>
<td>0.73</td>
<td>0.69</td>
</tr>
<tr>
<td>100%</td>
<td>3.25</td>
<td>3.40</td>
<td>3.12</td>
</tr>
<tr>
<td>120%</td>
<td>3.85</td>
<td>3.53</td>
<td>3.46</td>
</tr>
</tbody>
</table>

Table 1: Average Round-Trip Time of SIP Messages

The resources are reserved for traffic which is very susceptible to congestion and require a constant minimum bandwidth like audio or video traffic. This causes other traffic like SIP messages, which pass by classic best effort method, to suffer. This not only results in delaying of such packets but also causes some packet loss.

C. Processing Time by SIP Proxies

This was calculated by subtracting the time at which a message enters a proxy from the time at which it exits the proxy. As processing time of a QoS aware proxy and the one that is not aware of QoS should theoretically only differ when processing INVITE, OK or BYE messages, therefore processing time for only these messages was observed. Note that these processing times are calculated when the proxy is calculating only one message at a time. Table 2 depicts the observed trends.

<table>
<thead>
<tr>
<th>SIP Message</th>
<th>Standard SIP Proxy (sec)</th>
<th>QoS Aware SIP Proxy (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVITE</td>
<td>0.062</td>
<td>0.065</td>
</tr>
<tr>
<td>OK</td>
<td>0.027</td>
<td>0.029</td>
</tr>
<tr>
<td>BYE</td>
<td>0.046</td>
<td>0.052</td>
</tr>
</tbody>
</table>

Table 2: Average Processing Time of SIP Messages

It can be noticed that the INVITE and BYE take more time to get processed than the OK. This is due to the fact that INVITE and BYE are request messages that create transactions whereas OK being a response is always a part of an existing transaction. In any case, the processing time for both requests and response does not differ much between a QoS aware and a QoS unaware proxy.

V. Future Work

Further work can be done to include more functionality and thus enhance the techniques discussed here. The performance evaluation has shown that the functioning of the QoS aware SIP Proxy is quite satisfactory; however the transmission delay that SIP messages incur when network traffic increases should be minimized. This could be achieved by prioritizing the SIP messages or by having a dedicated service for the SIP signaling so that the user could have an unnoticeable post-pickup delay.

Another feature that can be added is the use of specific bandwidth for making QoS reservations. As the QoS server used is under construction, it is expected that it will include statistical admission control features to be able to benefit from the multiplexing of several VoIP connections. The SIP Proxy can know about the user’s bandwidth requirements by detecting the codecs it has offered to communicate with. Although this has already been implemented in the proxy, it requires a rigorous database of codecs and the bandwidth they usually require.

REFERENCES