Implementation, validation and practical verification of SIP QoS-aware application for the federated tactical systems

PIOTR ŁUBKOWSKI¹, DAMIAN DUDA²

¹ Military University of Technology, Faculty of Electronics, Kaliskiego 2, 00-908 Warsaw, Poland
² Military Communication Institute, ul. Warszawska 22A, 05-130 Zegrze Południowe, Poland

Abstract: The signalling layer is considered as a tool to create standardized interface to advanced capabilities of QoS-enabled network based on IP protocol. The Session Initiation Protocol (SIP) is chosen as a main signalling protocol for many research projects as well as standardization activities in the area of QoS IP networks. The authors of the paper adopted this concept and proposed the enhancement to the standard SIP protocol allowing explicit signalling of QoS requirements by QoS-aware terminals in the Federation of Systems. Such solution allows for more efficient use of network resources. The solution presented in the paper was implemented in the QoS demonstrator (DQoS) established in the MUT and MCI laboratories. The paper covers the verification and validation of the proposal by experimental studies as well as a discussion of some selected results of experiments.

Keywords: SIP extension, QoS-Aware application, Federation of Systems.

Introduction

Providing end-to-end Quality of Service for applications in the Federation of Systems (FoS) requires appropriate traffic control mechanisms to handle user traffic transferred through such systems. The requirement is mandatory in the tactical and battlefield level networks due to their properties like dynamic topology changes, mobility of users and service providers, common use of wireless links and limited processing and power capacity of network nodes.

The signalling is the mechanisms for providing exchange of control and management information in a QoS IP network [1]. It can also be seen as an interface to advanced capabilities of QoS-enabled network. The Session Initiation Protocol (SIP) is chosen as a main signalling protocol for many research projects.
as well as standardization activities in the area of QoS IP networks. The authors of the paper adopted this concept and proposed the enhancement to the standard SIP protocol allowing explicit signalling of QoS requirements by QoS-aware terminals in the Federation of Systems. The proposed enhancement includes a new SIP method called ORDER that is passed to control plane as explicit set of QoS parameters required by the session. The main reason for the development of an additional SIP method was that the capturing of INVITE message and analysis of SDP payload may not be feasible, if there are security mechanisms used to provide integrity and confidentiality of SIP messages in the end-to-end context. By explicit signalling of QoS descriptor the need to capture end-to-end signalling messages (e.g. INVITE with SDP payload) can be avoided. Thus an endpoint capable of submitting appropriate formatted signalling message can be granted access to features of QoS IP network. It may be useful in particular for handling traffic of web-services-based applications. We also expect that the proposal may help to better recognize QoS-aware terminals that can publish their abilities this way. The proposed concept was initially described in [2].

The QoS-Aware Session signaling

As it was already mentioned the explicit signalling of QoS requirements is important in the implementation of the QoS-aware services in the Federation of Systems' environment. The INVITE method starts the establishing of the SIP session. It contains information necessary to describe a session, included in the SDP protocol payload. The negotiations that are executed between the involved sides ends when the called terminal accepts the connection constraints. These constrains are necessary for QoS parameters reservation required in the QoS-aware session and are included in a new SIP method called ORDER that is passed to QoS Broker. The ORDER method is proposed as a way to provide a “user-network” interface to the QoS functions in the network. The QoS Broker is a part of QoS supporting components and consists of a SIG-Proxy and a Resource Manager with Admission Control functionality (Figure 1).

Figure 1. The QoS support components
A Resource Manager module is a QoS Policy Decision Point (PDP) which in the case of resource availability exchanges management messages with the Policy Enforcement Points (PEP) located in the edge devices (edge routers of the access networks). The resource reservation process is conducted according to adequate traffic conditioning agreement (QoS policy), configured in the Resource Manager. The QoS policy maintained in the Resource Manager is enforced by the PEP located in the edge device. Output interfaces of the router are configured using the composition of the Priority Queueing (PQ) and Hierarchical Token Bucket (HTB) mechanisms, depending on the aggregated class of service, with adequate Differentiated Service Code Point (DSCP).

If the Resource Manager module confirms the resource reservation process, the local SIG-Proxy forwards the ORDER message to the SIG-Proxy located in the destination network involved in the QoS session. It initiates the reservation process in that domain in the same way as before. When the process is successfully finished, the SIG-Proxy confirms the successful session establishment to the calling access network. The signalling diagram for the QoS-aware session establishment is presented in Figure 2.

The session termination is indicated by BYE method. It can be sent by each side of the existing connection. When the method BYE is received at the QoS Broker, it initiates the process of resource releasing in each of access domains that have been engaged in the SIP session.

![Diagram](Image)

Figure 2. The QoS-aware session signalling procedure
The method for explicit signalling of QoS requirements

A new ORDER method was chosen as the base for the implementation of QoS signalling functionality in the Federation of Systems’ environment. It was assumed that this method would be used on demand by the calling subscriber in the case of degradation of the voice connection’s perceived quality. This gives us the opportunity to aggregate the signalling traffic exchanged between the co-operating domains. In such case signalling messages are exchanged only among the signalling nodes of mentioned access domains.

The ORDER method is built on the standard format of a SIP method and consists of a request-line, a request-header and a body. The request-line as well as the request-header are the same as for the INVITE method defined in the SIP specification [3]. Only modifications of the request-line and the request-header concern the “Request-Line” line and the “Cseq:” line which contains the name of the ORDER method. The format of the ORDER method was described in [4] and it is shown in Figure 3.

The essential modifications were proposed for the message body, which is based on the SDP message format defined in RFC 4566 [5] and described in [4]. The message body of the ORDER method includes parameters for the session recognition exchanged between both sides of the connection, as well as a few new parameters which are used for the QoS-aware session description. The QoS parameters agreed during the phase of session establishment are contained in the “m” line of the SDP message. We assumed that the message body of the ORDER method would contain a two “m” lines – one for the calling and one for the called side. It gives us the opportunity to recognize the sort of media types, protocols and ports for which the reservation should be executed. In our proposal we assume that the “b” line (bandwidth line) should be also used. This line contains the value of a bandwidth, which should be reserved by an appropriate QoS Broker in every access domain involved in the QoS-aware session.

Figure 3. The format of the new ORDER method with the modifications in Request-Line and CSeq lines
For each direction we propose to introduce a so-called QoS Description line marked with the “q” index. We presumed that this line should look as follows:

\[ q= <\text{CoS}> <\text{ToR}> [<\text{qtpmap } 0> <\text{qtpmap1}> \ldots] \]

where:

<CoS> – Class of Service according to the DiffServ IETF specification;
<ToR> – Type of Reservation;
<qtpmap0 \ldots> – mapping of QoS parameters to values

The allowed values for particular fields are described in Table 1:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Recognized value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CoS</td>
<td>telephone</td>
</tr>
<tr>
<td></td>
<td>mmstreaming</td>
</tr>
<tr>
<td></td>
<td>bcvideo</td>
</tr>
<tr>
<td></td>
<td>lldata</td>
</tr>
<tr>
<td></td>
<td>standard</td>
</tr>
<tr>
<td>ToR</td>
<td>resv</td>
</tr>
<tr>
<td></td>
<td>prio</td>
</tr>
<tr>
<td>qtpmap0..qtpmap9</td>
<td>0.9</td>
</tr>
</tbody>
</table>

In implemented software, a “q” line with the <CoS> and <ToR> fields is used. The message body of the ORDER method with proposed modifications is presented in Figure 4.

![Session Description Protocol](image)

Figure 4. The message body of the ORDER method

In order to acknowledge or refuse the acceptance of the QoS-aware session establishment, a set of responses for the SIG-Proxy is included in our enhancements to SIP signalling protocol. They are presented below in Table 2.
Table 2. The possible responses to the ORDER method

<table>
<thead>
<tr>
<th>Type of response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200 OK</td>
<td>Acceptance of the ORDER request</td>
</tr>
<tr>
<td>600 Busy Everywhere</td>
<td>Lack of free resources for the ORDER request</td>
</tr>
<tr>
<td>503 Service Unavailable</td>
<td>The temporal unavailability of service, other than lack or resources</td>
</tr>
<tr>
<td>500 Server Internal Error</td>
<td>Error in the communication with the Router</td>
</tr>
</tbody>
</table>

The examples of “200 OK” and “Busy Everywhere” responses are shown in Figure 5.

Figure 5. The examples of the SIG-Proxy responses to the ORDER method

Practical implementation of the order method

The enhancements mentioned above are implemented in the form of a QoS-aware terminal and the SIG-Proxy applications. These applications together with RM and AC are implemented in the QoS demonstrator (DQoS). The functionality of RM and AC modules is outside the scope of this paper.

A QoS-aware terminal application is called **nec_kphone** and is based on the Kphone SIP UA (User Agent) for Linux. It is licensed under the GNU General Public License and written in C++ language. Kphone enables session establishing between two endpoints and supports as well SIP Proxy Agents as means of direct communication between User Agents. The available session types are as follows:

- audio;
- instant messaging;
- video (with the external application).

The implementation of the ORDER method in the **nec_kphone** software environment was performed in reference to kcallwidget, sipcall, siptransaction and sipprotocol program modules. The following functions were developed for this implementation:

- **requestOrder** which prepares the content of the ORDER method header and body;
- **sendRequestOrder** which is responsible for sending the ORDER request to the SIG-Proxy;
- **handlingOrderResponse** which is responsible for handling responses to the ORDER method.

An `ordermsg` function which creates a message body for the ORDER request was developed in the `sdp` program module. The content of this function is presented in Figure 6.

The `nec_kphone` application window during the time of connection establishment phase is shown in Figure 7. A connection is established when the “Połąc” option is selected. The SIP INVITE message is send to the remote VoIP subscriber in order to signal a new connection. This message is transferred through a SIP Proxy that is a part of the QoS concept presented in [2]. As soon as the connection is established the subscriber has a possibility to use a new SIP ORDER method for the assured service demand servicing. It is available by selecting the “QoS Order” option. After that the ORDER message is transferred to the SIP Proxy, which initiates the procedure of resource reservation. This message is filled up with SDP payload that contains the requested DSCP and required bandwidth. If there are free resources available, an application demand is gratified.

```cpp
QString SdpMessage::ordermsg()
{
    QString input = neworder;
    QString ordermsg = input;
    QString curline;
    QString sendipaddress;
    QString codecstr;

    input += '\n';
    /*
    if( input.left( input.find( '\n' ) ) != "v=0" ) {
        return;
    }*/
    if( curline.left( curline.find( '=' ) ) == "c" ) {
        sendipaddress = curline.mid( curline.find("IP4") + 4 ).stripWhiteSpace();
    }
    if( curline.left( curline.find("=" ) ) == "m" ) {
        codecstr = curline.mid( curline.find("RTP/AVP") + 7 );
        codecstr += " ";
    }

    // Connection Info
    ordermsg += "c=IN IP4 " + sendipaddress + "\r\n";
    // Required Bandwidth
    ordermsg += "b=AS1000000\r\n";
    // QoS class
    ordermsg += "q=telephony resv\r\n";
    // Body for called part
    ordermsg += "m=audio " + QString::number( reported ) + " RTP/AVP " + reccodecstr + "\r\n";
    ordermsg += "c=IN IP4 " + recipaddress + "\r\n";
    // Required Bandwidth
    ordermsg += "b=AS1000000\r\n";
    // QoS class
    ordermsg += "q=telephony resv\r\n";
    return ordermsg;
}
```

Figure 6. The content of the `ordermsg` function
The SIG-Proxy application is based on the SIP Express Router (SER) version 0.9.6a. This is a high-performance, configurable, free SIP server licensed under the open-source GNU General Public License. It can act as a SIP (RFC 3261) registrar, proxy or redirect server. SER can be configured to serve specialized purposes such as load balancing or SIP front-end to application servers [8].

The implementation of the SIP signalling enhancements in reference to the SIG-Proxy application triggers a modification of the SER software. The following modifications were introduced:

- a maintenance of the ORDER method;
- a communication with a Resource Manager module of the QoS Broker;
- a transfer and control of signalling messages exchanged in the local domain as well as among autonomous systems.

These modifications create new functionalities of the SIG-Proxy application related to the QoS supporting in the inter-domain communication. They are presented below in the form of a block diagram presenting the flow of the ORDER message in the SIG-Proxy application (Figure 8).
Figure 8. The flow of the ORDER message in the SIG-Proxy application
The SIG-Proxy application window in the state of receiving the ORDER request is presented in Figure 9.

![Image of SIG-Proxy application window]

Figure 9. The SIG-Proxy application – receiving the ORDER request

Detailed description of the nec_kphone application and the SIG-Proxy application can be found in [6] and [7].

Validation and verification of the QoS-aware application

A. Experiment environment

All components of the QoS platform were implemented in the DQoS established in the MUT and MCI laboratories. The overall architecture of the QoS demonstrator is presented in Figure 10.

![Image of network architecture]

Figure 10. Network architecture for the QoS-Aware application testing
B. Assumptions

A set of experiments and scenarios proposed for verification and validation of the developed QoS enhancements is presented in Table 3.

Table 3. Practical experiments scenarios

<table>
<thead>
<tr>
<th>#</th>
<th>Experiment</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Functional tests of nec_kphone</td>
<td>Evaluation of nec_kphone functionality in relation to the ORDER method handling.</td>
</tr>
<tr>
<td></td>
<td>application</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Functional tests of the SIG Proxy</td>
<td>Specification of the correctness of modified SIP protocol messages handled in the SIG Proxy module.</td>
</tr>
<tr>
<td></td>
<td>signalling module</td>
<td></td>
</tr>
</tbody>
</table>

We defined three types of applications:

- A voice application that uses the Telephony CoS (Class of Service);
- A critical messaging application is mapped to Low-Latency Data CoS;
- And a control application mapped to Control CoS.

The applications can operate in two modes, i.e. with and without the modified SIP signalling.

The QoS requirements are encoded in the payload of the SIP ORDER message as described in previous chapter.

The ORDER message is sent by the end-user application to the SIG-Proxy that extracts required QoS parameters and sends them to the Resource Manager (RM). RM works out the decision about the acceptance or rejection of the request according to the actual link load and the QoS requirements. The Admission Control module is called, which performs the peak rate allocation algorithm. The result from AC is sent back to the SIG-Proxy. If the request is accepted the SIG-Proxy sends the ORDER to the next SIG-Proxy on the path toward destination or to the end terminal. The next-hop SIG-Proxy exchanges appropriate messages with the corresponding Resource Manager. The rejection of a request triggers sending the SIP Busy Everywhere response to the initiating end-point. If necessary, a management message with a release request is then sent to the Resource Manager.

C. Results and discussion

Figure 11 shows the process of signalling ORDER request in the nec_kphone application. As it is shown the order and content of signalling messages are consistent with the requirements for the ORDER method defined in this paper, as well as with the SIP specification described in [3].

The results of receiving the ORDER request in the SIG-Proxy application are presented in Figure 12 and Figure 13. It can be seen that after the receipt of the ORDER method, a SIG-Proxy application sends the “100 Trying” response to the nec_kphone application to signal that request is executed. Simultaneously,
the ORDER message is transferred to the Resource Manager (ZZ2) of the calling party domain and to the Resource Manager (ZZ1) of the called party domain. When the resources are available, the response “200 OK” is received and returned to the **nec_kphone** application. The content of the ORDER request registered in the SIG-Proxy application is consistent with the requirements defined for the ORDER method.

Figure 11. The process of signalling the ORDER request registered in the **nec_kphone** application

Figure 12. The process of signalling the ORDER request registered in the **SIG-Proxy** application

Functional tests of the SIG Proxy signalling module incorporate also the correctness of exchanging the QoS requirements with Resource Manager and Admission Control modules. It particularly concerns the appropriate handling of the ORDER message in RM, as well as configuring the accurate reservation parameters in the HTB mechanism of an access router in the cooperating domains. Figure 14 shows the ORDER message reception in AC module of a calling domain.
Figure 13. The content of the ORDER request registered in the SIG-Proxy application

Figure 14. Receiving the ORDER message in AC module of a calling side
This message was transferred from the SIG-Proxy application after receiving the QoS order from the calling subscriber. Upon the receipt of the ORDER message, the AC module is decreasing overall bandwidth assigned to the “telephony” class of service. The result of this action is presented in Figure 15.

At the same time, the Resource Manager configures the HTB classifier to handle the bandwidth reserved by the ORDER method. This is shown in Figure 16.
Summary

This paper presents the implementation of extensions to the SIP protocol. The modified protocol allows for an explicit transfer of QoS requirements to the control plane of IP QoS testbed. By taking into account the result of the connection negotiations, it is possible to utilise network resources more efficiently.

However, the modifications of end-points and network devices are necessary to achieve the required functionality. The "QoS-awareness" needs an extension of SIP messages generated by user and server agents. This was achieved by the implementation of the new ORDER method in the nec_kphone, as well as in the SIG-Proxy applications. The results of functional testing show that the proposed SIP enhancement implementation work correctly and can be used for resources’ reservation in selected services, if the network performance is degraded, e.g. due to overload, changes of topology, malfunction or damage of the network infrastructure. It is worth noting that it is possible only if all QoS supporting components are available.

The explicit inclusion of QoS requirements in the signalling message may help to manage some security issues like end-to-end encryption of signalling messages, but it needs further research.

It should be noted that the functionality of the QoS network requires a dynamic and run-time reconfiguration of network devices. This process is considerably time-consuming. The modification or elimination of this issue is a direction for further research and implementation work.

REFERENCES