Session Initiation Protocol Application Layer Gateway

Jurica Mikulić, Ivan Andrijić, Viktor Matić
Faculty of Electrical Engineering and Computing, Department of Telecommunications
Zagreb, Croatia
E-mail: viktor@tel.fer.hr

Abstract: **Since the transition from IPv4 to IPv6 is expected to be graduate, IPv4 and IPv6 networks will have to coexist and interoperate for some time to come. There are a few proposed transition mechanisms, one of which is Network Address Translation - Protocol Translation (NAT-PT).**

One of the main drawbacks of this approach is a need for an Application Layer Gateway (ALG) for each application layer protocol which carries address information (e.g. DNS and FTP). Since Session Initiation Protocol (SIP) is being hailed as the core protocol for multimedia communications in next generation networks [1], a SIP ALG should also be implemented and used in conjunction with NAT-PT.

1. INTRODUCTION

With the NAT-PT approach a gateway is placed between IPv4 and IPv6 networks. This gateway acts similar to Network Address Translator (NAT) [2]. The main advantage of this approach is that end devices and networks need only to run either IPv4 or IPv6. However, it breaks the end-to-end transparency of the Internet and is accompanied by similar problems as introduced by the use of NAT [3]; since NAT-PT mangles packets only on the IP layer, special attention needs to be paid to protocols which carry addressing information in the protocol messages themselves (e.g. SIP).

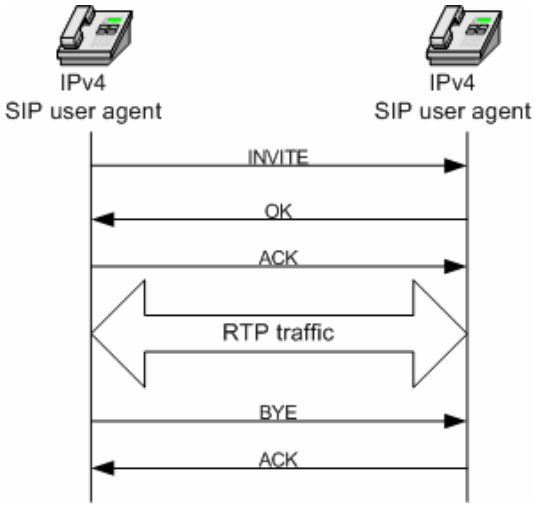

Figure 1 - SIP call flow in a homogeneous environment

A basic SIP call flow in a homogeneous network is shown in Figure 1. Note that even though this figure shows two IPv4 user agents (UA), this call flow can also be generated between two IPv6 SIP user agents. There are three main parts of a session: session establishment, Real Time Transport Protocol (RTP) traffic and session ending. Since addressing information is exchanged during session establishment, failure to correctly interpret this information on either side will lead to inability to establish the call. Figure 2 shows precisely that: in a heterogeneous environments, IPv4 addresses contained in SIP messages from the UA in IPv4 network will make no sense to the UA in IPv6-only environment (and vice versa). Hence, the media streaming can not be established.

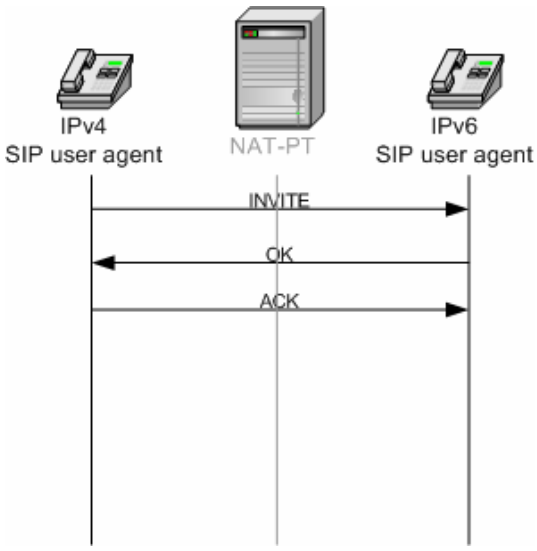

Figure 2 - SIP call flow in a heterogeneous environment

An additional protocol specific mechanism should therefore be implemented for SIP, as well as for each of those protocols which carry addressing information in the application layer. This mechanism is called the Application Layer Gateway (ALG). This paper analyzes problems concerning implementing a SIP ALG, and proposes an implementation solution.

We start out by elaborating basic principles which should be observed when implementing a SIP ALG. In section 3 we

analyze relevant SIP headers and SDP fields and elaborate their purpose. Section 4 proposes a solution for the SIP ALG implementation, while section 5 describes proposed testing scenarios for the implemented SIP ALG.

2. BASIC PRINCIPLES

Some basic principles should be observed when implementing a SIP ALG.

First of all, wherever it is possible, it is imperative to use Fully Qualified Domain Names (FQDN) of hosts instead of their IP addresses. Since the content of SIP messages depends exclusively on SIP User Agents (UA) and SIP proxy servers, this means that they should be configured to use FQDN wherever it is possible. With this approach hosts rely on DNS for address resolution, which assures up-to-date information about addresses which are dynamically mapped on the NAT-PT. Needless to say, this requires DNS servers in both IPv4 and IPv6 networks to be properly configured, and the NAT-PT to work in conjunction with a DNS ALG.

Furthermore, every SIP message can be mangled without knowledge of its context; SIP ALG does not need nor contain any information about the session a given SIP message is a part of. Therefore, SIP ALG can be stateless.

Finally, since information about all active address mappings is stored in the NAT-PT, a mechanism must be implemented which will allow the SIP ALG to access that information. In addition to that, NAT-PT must also be able to create additional mappings on request from the SIP ALG.

3. ANALYSIS OF RELEVANT SIP HEADERS AND SDP FIELDS

Session parameters are negotiated between SIP user agents during session initiation. Session Description Protocol (SDP) is used to convey these parameters between user agents. Since these parameters include address information, SDP content of SIP messages should also be inspected for IP addresses.

Only two SDP fields are of interest to the SIP ALG: origin field and connection field.

The origin field ("o=") gives the originator of the session (their username and the address of the user's host), plus a session ID and session version number. Address of the user's host is either the FQDN, or the IP address of the machine. For both IP4 and IP6, the FQDN is the form that should be given unless this is unavailable, in which case the globally unique address may be substituted [4].

The connection field ("c=") contains connection address. For unicast addresses, the connection address contains the fully qualified domain name or the unicast IP address of the expected data source, data relay or data sink, as determined by additional attribute fields. Furthermore, RFC 2327 [4] explicitly states that, if a unicast data stream is to pass through a NAT, the use of a fully-qualified domain name rather than an unicast IP address is recommended. One can therefore safely conclude that the use of FQDN is also recommended in messages traversing NAT-PT.

Note that RFC 2327 [4] clearly proposes use of FQDN in both the origin and the connection field. However, which address format is used depends exclusively on the implementation of the SIP user agent. These SDP fields must therefore be inspected for IP addresses in the SIP ALG.

In addition to these SDP fields, SIP ALG must also inspect and mangle two SIP headers: topmost Via header and Content-Length header.

The Via header field indicates the transport used for the transaction and identifies the location where the response is to be sent. Each proxy which forwards a SIP request adds its Via header to the message. Likewise, each proxy which forwards a SIP response removes its Via header from the message. Via header populating and processing is described in great detail in RFC 3261 [5].

The Content-Length header field indicates the size of the message body, in decimal number of octets, sent to the recipient. This header obviously can not contain any IP addresses. However, if the message content (SDP fields) is mangled, it is necessary to update this header after content mangling is complete.

4. PROPOSED SOLUTION

As we described in previous section, SIP ALG must inspect two SDP fields (origin and connection field) and two SIP headers (topmost Via header and Content-Length header). Upon reception of a SIP message, NAT-PT should therefore strip it of its IP headers and send it to the SIP ALG. After the message is returned from the ALG, NAT-PT should add new IP headers and forward the message. Since NAT-PT does not inspect the application layer data of packets it receives, SIP traffic can be recognized by the port it uses; standard port for SIP is 5060.

Three communication methods between the NAT-PT and the SIP ALG were considered: Java Native Interface (JNI), UDP sockets and TCP sockets. Use of JNI greatly simplifies coding but, because of the lack of debugging tools and JNI documentation, building even a basic interface between the NAT-PT and the SIP ALG can be very challenging. Use of either UDP or TCP sockets therefore seems like a better solution. Note that, if sockets are used, additional serialization mechanisms must be implemented, because all data must be serialized before it is transferred through sockets. Because of the blocking mechanisms already integrated in UDP, we propose the use of UDP sockets for

communication between the NAT-PT and the SIP ALG. This proposed solution can be seen in Figure 3.

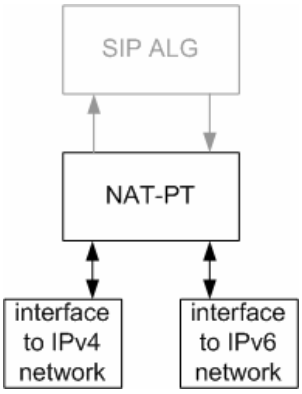

Figure 3 - Architecture used for SIP message mangling

Even though origin and connection field may contain different addresses, in basic call examples they contain the same address. In either case, mangling rules are identical for these two fields. First of all, addresses are checked to determine if they are FQDN or IP addresses. FQDN addresses can remain unchanged, but IP addresses must be mangled according to the following rules: if the message came from the IPv4 network and contains IPv4 address, the 96-bit prefix must be added, turning it into an IPv6 address. If the message came from the IPv6 network and contains IPv6 address, it must be replaced with the IPv4 address NAT-PT has mapped it to.

Topmost Via header must also be checked to see if it contains an IP address. If so, depending on the network this message came from (IPv4 or IPv6) and on whether it is a request or a response, it is mangled using the following rules:

	Request	Response
from IPv4	add the 96-bit prefix to the IPv4 address, turning it into an IPv6 address	replace the given IPv4 address with the IPv6 address NAT-PT has mapped it to
from IPv6	replace the given IPv6 address with the IPv4 address NAT-PT has mapped it to	remove the 96-bit prefix from the IPv6 address, turning it into an IPv4 address.

Table 1 - Via header mangling rules

If the message content (i.e. SDP part of the message) length has been altered, it is important to re-calculate it and update the Content-Length header of the SIP message. It is imperative that this is done AFTER the mangling of the message content (if any) is finished. This is the reason why SDP part of the message should be mangled before SIP headers.

Here is an example of the overall effect of the SIP ALG. These are the actual messages generated in testing environment. Headers and fields which the SIP ALG inspects are shown in bold.

This is an INVITE message with a SDP content, received by the NAT-PT from the IPv6 network:

```
INVITE sip:ua@fer.hr:5060 SIP/2.0
Call-ID:
  9865e47ba7979731fcabb0d0d5ec989e@jura.tel.fe
  r.hr
CSeq: 1 INVITE
From: "Jura @ IPv6"
  <sip:jura@tel.fer.hr:5060;transport=udp>;tag
  =17030800
To: <sip:ua@fer.hr:5060>
Via: SIP/2.0/UDP
  serverina.tel.fer.hr:5060;branch=z9hg4bkb2a1
  e359dcf1f40472bb67f1f654cfcd,SIP/2.0/UDP
  jura.tel.fer.hr:5060;branch=z9hG4bKa2a87d516
  1f87b6b9f3506fc2d26f732
Max-Forwards: 69
Contact: "Jura @ IPv6"
  <sip:jura.tel.fer.hr:5060;transport=udp>
Content-Type: application/sdp
Content-Length: 170

v=0
o=root 0 0 IN IP6 [fec0:0:0:0:0:0:0:20]
s=-
c=IN IP6 fec0:0:0:0:0:0:0:20
t=0 0
m=audio 22224 RTP/AVP 4 3 0 5 6 8 15
m=video 22222 RTP/AVP 34 26 31
a=recvonly
```

This is the same message after it is mangled in the SIP ALG. Again, inspected (and altered) headers and fields are shown in bold:

```
INVITE sip:ua@fer.hr:5060 SIP/2.0
Call-ID:
  9865e47ba7979731fcabb0d0d5ec989e@jura.tel.fe
  r.hr
CSeq: 1 INVITE
From: "Jura @ IPv6"
  <sip:jura@tel.fer.hr:5060;transport=udp>;tag
  =17030800
To: <sip:ua@fer.hr:5060>
Via: SIP/2.0/UDP
  serverina.tel.fer.hr:5060;branch=z9hg4bkb2a1
  e359dcf1f40472bb67f1f654cfcd,SIP/2.0/UDP
  jura.tel.fer.hr:5060;branch=z9hG4bKa2a87d516
  1f87b6b9f3506fc2d26f732
Max-Forwards: 69
Contact: "Jura @ IPv6"
  <sip:jura.tel.fer.hr:5060;transport=udp>
Content-Type: application/sdp
Content-Length: 154

v=0
o=root 0 0 IN IP4 161.53.19.40
s=-
c=IN IP4 161.53.19.40
```

```
t=0 0
m=audio 22224 RTP/AVP 4 3 0 5 6 8 15
m=video 22222 RTP/AVP 34 26 31
a=recvonly
```

5. TESTING SCENARIOS

Once implemented, the SIP ALG should be tested in three possible network configurations:

1. One SIP proxy in each network
2. Only one SIP proxy, located in IPv4 network
3. Only one SIP proxy, located in IPv6 network

In the first network configuration each SIP user agent registers with the proxy server in its network. In the remaining two network configurations both SIP user agents register with the only existing SIP proxy. Note that SIP proxy server also implements functionality of a SIP registrar server.

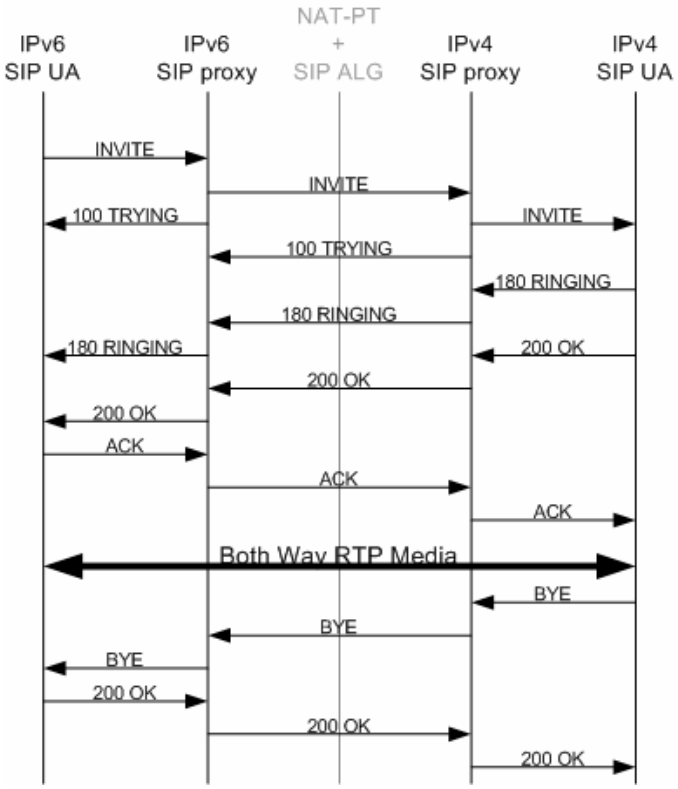

Figure 4 - SIP call flow over a NAT-PT with a SIP ALG.

For each of these network configurations, depending on which user agent initiates the call and which one ends it, there are four possible testing scenarios. This gives a total of twelve possible testing scenarios, in which the SIP ALG must enable session initiation, session maintaining and session ending. One of these scenarios, in which the UA from the IPv6 network initiates the call and the UA from the IPv4 network ends it, is shown in Figure 4. In this particular scenario there is a SIP proxy in each network.

The SIP ALG implemented according to solutions proposed in this paper was successfully tested in all twelve of these scenarios.

6. CONCLUSION

As major transition mechanisms to IPv6 we can distinguish three different approaches: dual stacks, tunneling and protocol translators.

The dual stack approach presumes that the networks run both IPv4 and IPv6 routing protocols, and the end systems are capable of sending and receiving both IPv4 and IPv6 packets. While this is a simple transition mechanism, it requires providing IPv4 addresses to all end systems, which negates the major advantage of IPv6. Furthermore, it complicates the network architecture, as it requires managing both IPv4 and IPv6 routing protocols.

With the tunneling approach IPv6 islands are connected through tunnels established over IPv4 networks. While simple to deploy in restricted areas, managing a large number of tunnels becomes complicated. Furthermore, this mechanism does not enable IPv4-only and IPv6-only end systems to communicate.

With the NAT-PT approach a gateway, acting similar to Network Address Translators (NAT), is placed between IPv4 and IPv6 networks. This approach enables IPv4-only and IPv6-only end systems to interoperate, but is accompanied by similar problems as introduced by the use of NAT, and special attention therefore needs to be paid to application layer protocols which carry addressing information in the protocol messages themselves. An Application Layer Gateway (ALG) must be implemented for each of these protocols.

We believe that NAT-PT is by no means a perfect transition mechanism. However, when it comes to communication and interoperability of IPv4-only and IPv6-only end systems, it appears to have no real alternative.

REFERENCES

[1] Dorgham Sisalem, Jens Fiedler: "SIP and IPv6: Why and How?", Fraunhofer Institute FOKUS, 2002.

[2] P. Srisuresh, M. Holdrege: "RFC 2663: IP Network Address Translator (NAT) Terminology and Considerations", IETF, 1999.

[3] M. Holdrege, P. Srisuresh: "RFC 3027: Protocol Complications with the IP Network Address Translator", IETF, 2001.

[4] M. Handley, V. Jacobson: "RFC 2327: SDP: Session Description Protocol", IETF, 1998.

[5] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler:"RFC 3261: SIP: Session Initiation Protocol", IETF., 2002.