SIPPING WG   V. Gurbani, Ed.
Internet-Draft   Bell Laboratories, Alcatel-Lucent
Intended status: Informational   C. Boulton
Expires: September 6, 2007   Ubiquity Software Corporation
                                                                R. Sparks
                                                        Estacado Systems
                                                           March 5, 2007

Session Initiation Protocol (SIP) Torture Test Messages for Internet Protocol Version 6 (IPv6)
draft-ietf-sipping-ipv6-torture-tests-01

Status of this Memo

Copyright Notice

Abstract

This informational document provides examples of Session Initiation Protocol (SIP) test messages designed to exercise and "torture" the code of a SIP implementation that parses IPv6 addresses.

This work is being discussed on the sipping@ietf.org mailing list.

Table of Contents

1. Overview

This document is informational, and is NOT NORMATIVE on any aspect of SIP.

This document contains test messages based on the current version (2.0) of the Session Initiation Protocol as defined in [RFC3261].

This document is expected to be used as a companion document to the more general SIP torture test document [RFC4475], which does not include specific tests for IPv6 network identifiers.

This document does not attempt to catalog every way to make an invalid message, nor does it attempt to be comprehensive in exploring unusual, but valid, messages. Instead, it tries to focus on areas that may cause interoperability problems in IPv6 deployments.

2. Document conventions

This document contains many example SIP messages. The appendix contains an encoded binary form containing the bit-exact representation of the messages and the algorithm needed to decode them into separate files.

The IPv6 addresses used in this document correspond to the 2001:DB8::/32 address prefix reserved for documentation [RFC3489]. Likewise, the IPv4 addresses used in this document correspond to the 192.0.2.0/24 address block as described in [RFC3330].

Although SIP is a text-based protocol, some of these examples cannot be unambiguously rendered without additional markup due to the constraints placed on the formatting of RFCs. This document uses the <allOneLine/> markup convention established in [RFC4475] to avoid ambiguity and meet the Internet-Draft layout requirements. For the sake of completeness, the text defining this markup from Section 2.1 of [RFC4475] is reproduced in its entirety below:

"Several of these examples contain unfolded lines longer than 72 characters. These are captured between <allOneLine/> tags. The single unfolded line is reconstructed by directly concatenating all lines appearing between the tags (discarding any line feeds or carriage returns). There will be no whitespace at the end of lines. Any whitespace appearing at a fold-point will appear at the beginning of a line.

"The following represent the same string of bits:

```
      Header-name: first value, reallylongsecondvalue, third value


         <allOneLine>
         Header-name: first value,
          reallylongsecondvalue
         , third value
         </allOneLine>

         <allOneLine>
         Header-name: first value,
          reallylong
         second
         value,
          third value
         </allOneLine>
```

"Note that this is NOT SIP header-line folding, where different strings of bits have equivalent meaning."

3. SIP and IPv6 network configuration

System-level issues like deploying a dual-stack proxy server, populating DNS with A and AAAA RRs, zero-configuration discovery of outbound proxies for IPv4 and IPv6 networks, when should a dual-stack proxy Record-Route itself, and media issues also play a major part in the transition to IPv6. This document does not, however, address these issues. Instead, a companion document [ID.sip-trans] provides more guidance on these.

4. Parser torture tests

The test messages are organized into several sections. Some stress only a SIP parser and others stress both the parser and the application above it. Some messages are valid, and some are not. Each example clearly calls out what makes any invalid messages incorrect.

Please refer to the ABNF in [RFC3261] on representing IPv6 references in SIP. IPv6 references are delimited by a "[" and "]". For Uniform Resource Identifiers (URI), RFC3261 mandates that the "IPv6reference" production rule be used when recognizing tokens that comprise an IPv6 reference. More specifically, the ABNF states:

```
     SIP-URI        =  "sip:" [ userinfo ] hostport
                       uri-parameters [ headers ]
     hostport       =  host [ ":" port ]
     host           =  hostname / IPv4address / IPv6reference
     IPv6reference  =  "[" IPv6address "]"
     IPv6address    =  hexpart [ ":" IPv4address ]
     hexpart        =  hexseq / hexseq "::" [ hexseq ] / "::" [ hexseq ]
     hexseq         =  hex4 *( ":" hex4)
     hex4           =  1*4HEXDIG
```

4.1. Valid SIP message with an IPv6 reference

The request below is well-formatted according to the grammar in RFC3261. An IPv6 reference appears in the Request-URI (R-URI), Via header, and Contact header.

Message Details: ipv6-good

```
REGISTER sip:[2001:db8::10] SIP/2.0
To: sip:user@example.com
From: sip:user@example.com;tag=81x2
Via: SIP/2.0/UDP [2001:db8::9:1];branch=z9hG4bKas3-111
Call-ID: SSG9559905523997077@hlau_4100
Max-Forwards: 70
Contact: "Caller" <sip:caller@[2001:db8::1]>
CSeq: 98176 REGISTER
Content-Length: 0
```

4.2. Invalid SIP message with an IPv6 reference

The request below is not well-formatted according to the grammar in RFC3261. The IPv6 reference in the R-URI does not contain the mandated delimiters for an IPv6 reference ("[" and "]").

An element receiving this request should respond with a 400 Bad Request error.

Message Details: ipv6-bad

```
REGISTER sip:2001:db8::10 SIP/2.0
To: sip:user@example.com
From: sip:user@example.com;tag=81x2
Via: SIP/2.0/UDP [2001:db8::9:1];branch=z9hG4bKas3-111
Call-ID: SSG9559905523997077@hlau_4100
Max-Forwards: 70
Contact: "Caller" <sip:caller@[2001:db8::1]>
CSeq: 98176 REGISTER
```

```
      Content-Length: 0
```

4.3. Port ambiguous in a URI

IPv6 uses the colon to delimit octets. This may lead to ambiguity if the port number on which to contact a SIP server is inadvertently conflated with the IPv6 reference. Consider the REGISTER request below. The sender of the request intended to specify a port number (5070) to contact a server, but inadvertently, put the port number inside the closing "]" of the IPv6 reference. Unfortunately, since the IPv6 address in the R-URI is compressed, the intended port number becomes the last octet of the reference.

From a parsing perspective, the request below is well-formed. However, from a semantic point of view, it will not yield the desired result. Implementations must take care to ensure that when a raw IPv6 address appears in a SIP URI, then any port number, if it is required, appears outside the closing "]" delimiting the IPv6 reference.

Message Details: port-ambiguous

```
      REGISTER sip:[2001:db8::10:5070] SIP/2.0
      To: sip:user@example.com
      From: sip:user@example.com;tag=81x2
      Via: SIP/2.0/UDP [2001:db8::9:1];branch=z9hG4bKas3-111
      Call-ID: SSG9559905523997077@hlau_4100
      Contact: "Caller" <sip:caller@[2001:db8::1]>
      Max-Forwards: 70
      CSeq: 98176 REGISTER
      Content-Length: 0
```

4.4. Port umabiguous in a URI

In contrast to the example in Section 4.3, the following REGISTER request leaves no ambiguity whatsoever on where the IPv6 address ends and the port number begins. This REGISTER request is well formatted per the grammar in RFC3261.

Message Details: port-umabiguous

```
      REGISTER sip:[2001:db8::10]:5070 SIP/2.0
      To: sip:user@example.com
      From: sip:user@example.com;tag=81x2
      Via: SIP/2.0/UDP [2001:db8::9:1];branch=z9hG4bKas3-111
      Call-ID: SSG9559905523997077@hlau_4100
      Contact: "Caller" <sip:caller@[2001:db8::1]>
      Max-Forwards: 70
      CSeq: 98176 REGISTER
      Content-Length: 0
```

4.5. IPv6 reference delimiters in Via header addresses

IPv6 references can also appear in Via headers; more specifically in the "sent-by" production rule and the "via-received" production rule. In the "sent-by" production rule, the sequence of octets comprising the IPv6 address is defined to appear as an "IPv6reference" non-terminal, thereby mandating the "[" and "]" delimiters. However, this is not the case for the "via-received" non-terminal. The "via-received" production rule is defined thusly:

```
      via-received = "received" EQUAL (IPv4address / IPv6address)
```

The "IPv6address" non-terminal is defined not to include the delimiting "[" and "]". This has lead to the situation documented during the 18th SIP Interoperability Event [Email-SIPit]:

> Those testing IPv6 made different assumptions about enclosing literal v6 addresses in Vias in []. By the end of the event, most implementations were accepting either. Its about 50/50 on what gets sent.

While it would be beneficial if the same non-terminal ("IPv6reference") was used for both the "sent-by" and "via-received" production rules, there has not been a consensus in the working group to that effect. Thus, the best that can be suggested is that implementations must follow the Robustness Principle [RFC1122] and be liberal in accepting a "received" parameter with or without the delimiting "[" and "]" tokens. When sending a request, implementations must not put the delimiting "[" and "]" tokens.

The two test cases below are designed to stress this behavior. An element receiving either of these messages must parse them successfully.

The request below contains an IPv6 address in the Via received parameter. The IPv6 address is delimited by "[" and "]". Even though this is not a valid request based on a strict interpretation

of the grammar in RFC3261, robust implementations must nonetheless be able to parse the topmost Via header and continue processing the request.

Message Details: param-1

```
BYE sip:[2001:db8::10] SIP/2.0
To: sip:user@example.com;tag=bd76ya
From: sip:user@example.com;tag=81x2
<allOneLine>
Via: SIP/2.0/UDP [2001:db8::9:1];received=[2001:db8::9:255];
branch=z9hG4bKas3-111
</allOneLine>
Call-ID: SSG9559905523997077@hlau_4100
Max-Forwards: 70
CSeq: 321 BYE
Content-Length: 0
```

The OPTIONS request below contains an IPv6 address in the Via received parameter without the adorning "[" and "]". This request is valid according to the grammar in RFC3261.

Message Details: param-2

```
OPTIONS sip:[2001:db8::10] SIP/2.0
To: sip:user@example.com
From: sip:user@example.com;tag=81x2
<allOneLine>
Via: SIP/2.0/UDP [2001:db8::9:1];received=2001:db8::9:255;
branch=z9hG4bKas3
</allOneLine>
Call-ID: SSG95523997077@hlau_4100
Max-Forwards: 70
Contact: "Caller" <sip:caller@[2001:db8::1]>
CSeq: 921 OPTIONS
Content-Length: 0
```

4.6. SIP request with IPv6 addresses in SDP body

This request below is valid and well-formed according to the grammar in RFC3261. Note that the IPv6 addresses in the SDP body do not have the delimiting "[" and "]".

Message Details: ipv6-in-sdp

```
  INVITE sip:user@[2001:db8::10] SIP/2.0
  To: sip:user@[2001:db8::10]
  From: sip:user@example.com;tag=81x2
  Via: SIP/2.0/UDP [2001:db8::9:1];branch=z9hG4bKas3-111
  Call-ID: SSG9559905523997077@hlau_4100
  Contact: "Caller" <sip:caller@[2001:db8::1]>
  CSeq: 8612 INVITE
  Max-Forwards: 70
  Content-Type: application/sdp
  Content-Length: 268

  v=0
  o=assistant 971731711378798081 0 IN IP6 2001:db8::20
  s=Live video feed for today's meeting
  c=IN IP6 2001:db8::1
  t=3338481189 3370017201
  m=audio 6000 RTP/AVP 2
  a=rtpmap:2 G726-32/8000
  m=video 6024 RTP/AVP 107
  a=rtpmap:107 H263-1998/90000
```

4.7. Multiple IP addresses in SIP headers

Th request below is valid and well-formed according to the grammar in RFC3261. The Via list contains a mix of IPv4 addresses and IPv6 references.

Message Details: mult-ip-in-header

```
  BYE sip:user@host.example.com SIP/2.0
  Via: SIP/2.0/UDP [2001:db8::9:1]:6050;branch=z9hG4bKas3-111
  Via: SIP/2.0/UDP 192.0.2.1;branch=z9hG4bKjhja8781hjuaij65144
  <allOneLine>
  Via: SIP/2.0/TCP [2001:db8::9:255];branch=z9hG4bK451jj;
  received=192.0.2.200
  </allOneLine>
  Call-ID: 997077@lau_4100
  Max-Forwards: 70
  CSeq: 89187 BYE
  To: sip:user@example.net;tag=9817--94
  From: sip:user@example.com;tag=81x2
  Content-Length: 0
```

4.8. Multiple IP addresses in SDP

The request below is valid and well-formed according to the grammar in RFC3261. The SDP contains multiple media lines, and each media line is identified by a different network connection address.

Message Details: mult-ip-in-sdp

```
INVITE sip:user@[2001:db8::10] SIP/2.0
To: sip:user@[2001:db8::10]
From: sip:user@example.com;tag=81x2
Via: SIP/2.0/UDP [2001:db8::9:1];branch=z9hG4bKas3-111
Call-ID: SSG9559905523997077@hlau_4100
Contact: "Caller" <sip:caller@[2001:db8::1]>
Max-Forwards: 70
CSeq: 8912 INVITE
Content-Type: application/sdp
Content-Length: 181

v=0
o=bob 280744730 28977631 IN IP4 host.example.com
s=
t=0 0
m=audio 22334 RTP/AVP 0
c=IN IP4 192.0.2.1
m=video 6024 RTP/AVP 107
c=IN IP6 2001:db8::1
a=rtpmap:107 H263-1998/90000
```

5. Security considerations

This document presents NON-NORMATIVE examples of SIP session establishment. The security considerations in [RFC3261] apply.

Parsers must carefully consider edge conditions and malicious input as part of their design. Attacks on many Internet systems use crafted input to cause implementations to behave in undesirable ways. Many of the messages in this draft are designed to stress a parser implementation at points traditionally used for such attacks. This document does not, however, attempt to be comprehensive. It contains some common pitfalls that the authors have discovered while parsing IPv6 identifiers in SIP implementations.

6. IANA considerations

This document does not contain any actions for IANA.

7. Acknowledgments

The authors thank Jeroen van Bemmel, Dennis Bijwaard, Gonzalo Camarillo, Bob Gilligan, Alan Jeffrey, Larry Kollasch, Erik Nordmark, Kumiko Ono, Pekka Pessi, and other members of the SIP-related working groups for input provided during the construction of the document and discussion of the test cases.

8. References

8.1. Normative references

[RFC1122] Braden, R., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, October 1989.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[RFC3330] IANA, "Special-Use IPv4 Addresses", RFC 3330, September 2002.

[RFC3489] Rosenberg, J., Weinberger, J., Huitema, C., and R. Mahy, "STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)", RFC 3489, March 2003.

[RFC4475] Sparks, R., Hawrylyshen, A., Johnston, A., Rosenberg, J., and H. Schulzrinne, "Session Initiation Protocol (SIP) Torture Test Messages", RFC 4475, May 2006.

8.2. Informative references

[ID.sip-trans]
Camarillo, G., El Malki, K., and V. Gurbani, "IPv6 Transition in the Session Initiation Protocol (SIP)", draft-ietf-sipping-v6-transition-04.txt (work in progress), September 2006.

[Email-SIPit]
Sparks, R., "preliminary report: SIPit 18", Electronic Mail archived at http://www1.ietf.org/mail-archive/web/sip/current/msg14103.html, April 2006.

Appendix A. Bit-exact archive of each test message

The following text block is an encoded, gzip compressed TAR archive of files that represent each of the example messages discussed in Section 4.

To recover the compressed archive file intact, the text of this document may be passed as input to the following Perl script (the output should be redirected to a file or piped to "tar -xzvf -").

```
#!/usr/bin/perl
use strict;
my $bdata = "";
use MIME::Base64;
while(<>) {
  if (/-- BEGIN MESSAGE ARCHIVE --/ .. /-- END MESSAGE ARCHIVE --/) {
       if ( m/^\s*[^\s]+\s*$/) {
           $bdata = $bdata . $_;
       }
  }
}
print decode_base64($bdata);
```

Alternatively, the base-64 encoded block can be edited by hand to remove document structure lines and fed as input to any base-64 decoding utility.

A.1. Encoded reference messages

```
     -- BEGIN MESSAGE ARCHIVE --
     H4sICFV46EUAA2ZpbGVzLnRhcgDtmV1zozYUhnPNr9DsTa+wdSSBgJRO2mw2
     9XS764ndzHQ6mY5stAbXfCxgN9lfvwLH2HESs9kGb7rm3BiMxAFZz3teS0Gy
     MPWR8I4aDAwYm4wdYYyBm7j4LA5vPzFmjB4BNjjBnAJw9T0QBqpdkw+1inmW
     i1SlXPwz2dkuzn2Z7ri+fJPVy/Hnfcjm4uLsvDcYnl2gLEgcokbe8UaW4wBG
     g16/SzpYG8ZOeXGeyfREXoswmcnOOA61N2kcPnzpOBcT14Jrol0GwlndqfvH
     6z76a53DduDqeJSKaOy7n2z/nI1+ExnVAUA7FbOZ3nutug7ObcOwbWwYhNo2
     x5yf+DMx/1vND6z9Lq71N3H6r0i9zEEca6dxlItx7qBXxR1k+gr9WDzfuDw5
     2cgNVz9ppwP50UG2peYkWg1DeQcZ5fpbGU1y30FY0771T9RoBAX/kzhuUgDq
     +Tcq/jEhJf+mQqjlv/m4w/8mIPiqVYADUICS/yDSMy9pLEcN/4A5VPzTEiGg
     2Gjr/z6i9+6yNzxbM/wlEnC3zQtQga+A3jKBoOXLPywihQQMbxLpIJEks2As
     8iCOugqTewJBTEvTFi7WYldkWaDmU5Qjm4MysxyAcovbFrYAYZUP9fomWj8P
     wVrmvg0WEi0CT8bog5Qe+hCnKI89cfNDhkIp8yCaaGP3Xl/QcpdSajELwLIR
     pbzAi2DQQlfMvSBGppqI6GLY7/582UdEE26aJ6FQNg+dc2LqlHQt1UI1XyY3
     MWFVcwXluoM6Qb8SU/0mtm117WKCf8+aeEgRzme5HiRFCfCl8HZK3NdGnf9T
     E2+t/9ws9J8RYrT6v4f45c8N8ffjLO9s6Hcl/3Xy7ZiqXj+i4ff6gq2OOqQD
     Wx2m/lRY3AJ/OhfB1DSAsbudh6dbiYlhbFcOZsB0epzKsVSq6rmrXKSoEqti
     cls/dpjIZYmwweJIDdDDBjiSeVnfCvuo6zb7ojr44szlBv+NWcA6/jlnFf+s
     bAfUUJ6w5b/5ODz/9zjtlSF8mv8DC1b+bxSPELEwZ4xTrI5szk0KS9/H0La8
     Ku+nPBxGuHJshKg/QJUHwyvbx9aa+bhbe9Ai1lq4RKQi1KHROVa7/kPW9Z/Q
     sv6rgWjr/z5iVf+fuPRTkj3yuHkjnof/qmLXFfj/vEBU0k4JlJX9xdXjfceS
     f9Jojlr+zfX6D+FsyT+jLf97iPf9Ye/9u8G3W/6twN/i/j7228g3tRyspOF2
     VA5AHpI4zXURjoLJPJ5nzeSo5d8gG/u/tPT/3Gj9/z7i8f0fx8D8/7EJ9Bz2
     /1A3gUr+51GjCvAk/svrwNRBy/8eYsf+bykALf/fN/9ttNHG4cZnWALKRAAq
     AAA=
     ====
     -- END MESSAGE ARCHIVE --
```

Authors' Addresses

Vijay K. Gurbani (editor)
Bell Laboratories, Alcatel-Lucent
2701 Lucent Lane
Rm 9F-546
Lisle, IL 60532
USA

Phone: +1 630 224 0216
Email: vkg@alcatel-lucent.com

Chris Boulton
Ubiquity Software Corporation
Building 3
West Fawr Lane
St Mellons
Cardiff, South Wales CF3 5EA

Email: cboulton@ubiquitysoftware.com

Robert J. Sparks
Estacado Systems

Email: RjS@estacado.net

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).