Network Working Group
Internet-Draft
Expires: September 16, 2005

R. Sparks, Ed.
Estacado Systems
A. Hawrylyshen
Jasomi Networks
A. Johnston
MCI
J. Rosenberg
Cisco Systems
H. Schulzrinne
Columbia University

March 18, 2005

Session Initiation Protocol Torture Test Messages
draft-ietf-sipping-torture-tests-05

Status of this Memo

Copyright Notice

Abstract

This informational document gives examples of Session Initiation Protocol (SIP) test messages designed to exercise and "torture" a SIP implementation.

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]. Some messages exercise SIP's use of SDP as described in [RFC3264].

These messages were developed and refined at the SIPIt interoperability test events.

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.

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 have caused interoperability problems or have particularly unfavorable characteristics if they are handled improperly. This document is a seed for a test plan, not a test plan in itself.

The messages are presented in the text using a set of markup conventions to avoid ambiguity and meet Internet-Draft layout requirements. To resolve any remaining ambiguity, a bit-accurate version of each message is encapsulated in an appendix.

2. Document Conventions

This document contains many example SIP messages. Although SIP is a text-based protocol, many of these examples cannot be unambiguously rendered without additional markup due to the constraints placed on the formatting of RFCs. This document defines and uses the markup defined in this section to remove that ambiguity. This markup uses the start and end tag conventions of XML, but does not define any XML document type.

The appendix contains an encoded binary form of all the messages and the algorithm needed to decode them into files.

2.1 Representing Long Lines

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.

2.2 Representing Non-printable Characters

Several examples contain binary message bodies or header field values containing non-ascii range UTF-8 encoded characters. These are rendered here as a pair of hexadecimal digits per octet between <hex/> tags. This rendering applies even inside quoted-strings.

The following represent the same string of bits:

```
Header-name: value one
```

```
Header-name: value<hex>206F6E</hex>e
```

The following is a Subject header field containing the euro symbol:

```
Subject: <hex>E282AC</hex>
```

2.3 Representing Long Repeating Strings

Several examples contain very large data values created with

repeating bit strings. Those will be rendered here using <repeat count=some_integer>value</repeat>. As with <hex> this rendering applies even inside quoted-strings.

For example, the value "abcabcabc" can be rendered as <repeat count=3>abc</repeat>. A display name of "1000000 bottles of beer" could be rendered as

```
To: "1<repeat count=6><hex>30</hex></repeat> bottles of beer"
    <sip:beer.example.com>
```

and a Max-Forwards header field with a value of one google will be rendered here as

```
Max-Forwards: 1<repeat count=100>0</repeat>
```

3. SIP Test Messages

3.1 Parser tests (syntax)

3.1.1 Valid messages

3.1.1.1 A short tortuous INVITE

This short, relatively human-readable message contains:

- line folding all over
- escaped characters within quotes
- an empty subject
- LWS between colons, semicolons, header field values, and other fields
- both comma separated and separate listing of header field values
- mix or short and long form for the same header field name
- unknown header fields
- unknown header field with a value that would be syntactically invalid if it were defined in terms of generic-param
- unusual header field ordering
- unusual header field name character case
- unknown parameters of a known header field
- uri parameter with no value
- header parameter with no value
- integer fields (Max-Forwards and CSeq) with leading zeros

All elements should treat this as a well-formed request.

The UnknownHeaderWithUnusualValue header field deserves special attention. If this header field were defined in terms of comma separated values with semicolon separated parameters (as many of the existing defined header fields), this would be invalid. However,

since the receiving element does not know the definition of the syntax for this field, it must parse it as a header-value. Proxies would forward this header field unchanged. Endpoints would ignore the header field.

```
Message Details : wsinv

INVITE sip:vivekg@chair-dnrc.example.com;unknownparam SIP/2.0
TO :
 sip:vivekg@chair-dnrc.example.com ;   tag    = 1918181833n
from   : "J Rosenberg \\\""       <sip:jdrosen@example.com>
  ;
  tag = 98asjd8
MaX-fOrWaRdS: 0068
Call-ID: wsinv.ndaksdj@192.0.2.1
Content-Length   : 151
cseq: 0009
  INVITE
Via  : SIP  /   2.0
 /UDP
    192.0.2.2;branch=390skdjuw
s :
NewFangledHeader:   newfangled value
 continued newfangled value
UnknownHeaderWithUnusualValue: ;;,,;;,;
Content-Type: application/sdp
Route:
 <sip:services.example.com;lr;unknownwith=value;unknown-no-value>
v:  SIP  / 2.0  / TCP     spindle.example.com   ;
  branch  =   z9hG4bK9ikj8  ,
 SIP  /    2.0   / UDP  192.168.255.111   ; branch=
 z9hG4bK30239
m:"Quoted string \"\"" <sip:jdrosen@example.com> ; newparam =
      newvalue ;
  secondparam ; q = 0.33

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.3
s=-
c=IN IP4 192.0.2.4
t=0 0
m=audio 492170 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.1.2 Wide range of valid characters

This message exercises a wider range of characters in several key syntactic elements than implementations usually see. Of particular note:

- The Method contains non-alpha characters from token. Note that % is not an escape character for this field. A method of IN%56ITE is an unknown method. It is not the same as a method of INVITE
- The Request-URI contain unusual, but legal, characters
- A branch parameter contains all non-alphanum characters from token
- The To header field value's quoted-string contains quoted-pair expansions, including a quoted NULL character
- The name part of name-addr in the From header field value contains multiple tokens (instead of a quoted string) with all non-alphanum characters from the token production rule. That value also has an unknown header parameter whose name contains the non-alphanum token characters and whose value is a non-ascii range UTF-8 encoded string. The tag parameter on this value contains the non-alphanum token characters
- The Call-ID header field value contains the non-alphanum characters from word. Notice that in this production:
  - % is not an escape character. (It is only an escape character in productions matching the rule "escaped")
  - " does not start a quoted-string. None of ',` or " imply that there will be a matching symbol later in the string
  - The characters []{}()<> do not have any grouping semantics. They are not required to appear in balanced pairs
- There is an unknown header field (matching extension-header) with non-alphanum token characters in its name and a UTF8-NONASCII value

If this unusual URI has been defined at a proxy, the proxy will forward this request normally. Otherwise a proxy will generate a 404. Endpoints will generate a 501 listing the methods they understand in an Allow header field.

```
Message Details : intmeth

<allOneLine>
!interesting-Method0123456789_*+`.%indeed'~
 sip:1_unusual.URI~(to-be!sure)&isn't+it$/crazy?,/;;*
:&it+has=1,weird!*pas$wo~d_too.(doesn't-it)
@example.com SIP/2.0
</allOneLine>
Via: SIP/2.0/TCP host1.example.com;branch=z9hG4bK-.!%66*_+`'~
<allOneLine>
To: "BEL:\<hex>07</hex> NUL:\<hex>00</hex> DEL:\<hex>7F</hex>"
 <sip:1_unusual.URI~(to-be!sure)&isn't+it$/crazy?,/;;*
@example.com>
</allOneLine>
<allOneLine>
From: token1~` token2'+_ token3*%!.- <sip:mundane@example.com>
;fromParam''~+*_!.-%=
"<hex>D180D0B0D0B1D0BED182D0B0D18ED189D0B8D0B9</hex>"
;tag=_token~1'+`*%!-.
</allOneLine>
Call-ID: intmeth.word%ZK-!.*_+'@word`~)(><:\/"][?}{
CSeq: 139122385 !interesting-Method0123456789_*+`.%indeed'~
Max-Forwards: 255
<allOneLine>
extensionHeader-!.%*+_`'~:
<hex>EFBBBFE5A4A7E5819CE99BBB</hex>
</allOneLine>
Content-Length: 0
```

3.1.1.3 Valid use of the % escaping mechanism

This INVITE exercises the % HEX HEX escaping mechanism in several places. The request is syntactically valid. Interesting features include:

- The request-URI has sips:user@example.com embedded in its userpart. What that might mean to example.net is beyond the scope of this document.
- The From and To URIs have escaped characters in their userparts.
- The Contact URI has escaped characters in the URI parameters. Note that the "name" uri-parameter has a value of "value%41" which is NOT equivalent to "valueA". Per [RFC2396], unescaping URI components is never performed recursively.

A parser must accept this as a well-formed message. The application using the message must treat the % HEX HEX expansions as equivalent to the character being encoded. The application must not try to

interpret % as an escape character in those places where % HEX HEX ("escaped" in the grammar) is not a valid part of the construction. In [RFC3261], "escaped" only occurs in the expansions of SIP-URI, SIPS-URI, and Reason-Phrase

Message Details : esc01

```
INVITE sip:sips%3Auser%40example.com@example.net SIP/2.0
To: sip:%75se%72@example.com
From: <sip:I%20have%20spaces@example.net>;tag=938
Max-Forwards: 87
i: esc01.239409asdfakjkn23onasd0-3234
CSeq: 234234 INVITE
Via: SIP/2.0/UDP host5.example.net;branch=z9hG4bKkdjuw
C: application/sdp
Contact:
  <sip:cal%6Cer@host5.example.net;%6C%72;n%61me=v%61lue%25%34%31>
Content-Length: 151

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.1
s=-
c=IN IP4 192.0.2.1
t=0 0
m=audio 492170 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.1.4 Escaped nulls in URIs

This register request contains several URIs with nulls in the userpart. The message is well formed - parsers must accept this message. Implementations must take special care when unescaping the AOR in this request to not prematurely shorten the username. This request registers two distinct contact URIs.

Message Details : escnull

```
REGISTER sip:example.com SIP/2.0
To: sip:null-%00-null@example.com
From: sip:null-%00-null@example.com;tag=839923423
Max-Forwards: 70
Call-ID: escnull.39203ndfvkjdasfkq3w4otrq0adsfdfnavd
CSeq: 14398234 REGISTER
Via: SIP/2.0/UDP host5.example.com;branch=z9hG4bKkdjuw
Contact: <sip:%00@host5.example.com>
Contact: <sip:%00%00@host5.example.com>
```

```
L:0
```

3.1.1.5 Use of % when it is not an escape

Most of the places % can appear in a SIP message, it is not an escape character. This can surprise the unwary implementor. The following well-formed request has these properties:

- The request method is unknown. It is NOT equivalent to REGISTER
- The display-name portion of the To and From header fields is "%Z%45". Note that this is not the same as %ZE
- This message has two Contact header field values, not three. %lt;sip:alias2@host2.example.com%gt; is a C%6Fntact header field value

A parser should accept this message as well formed. A proxy would forward or reject the message depending on what the Request-URI meant to it. An endpoint would reject this message with a 501.

```
Message Details : esc02

RE%47IST%45R sip:registrar.example.com SIP/2.0
To: "%Z%45" <sip:resource@example.com>
From: "%Z%45" <sip:resource@example.com>;tag=f232jadfj23
Call-ID: esc02.asdfnqwo34rq23i34jrjasdcnl23nrlknsdf
Via: SIP/2.0/TCP host.example.com;branch=z9hG4bK209%fzsnel234
CSeq: 29344 RE%47IST%45R
Max-Forwards: 70
Contact: <sip:alias1@host1.example.com>
C%6Fntact: <sip:alias2@host2.example.com>
Contact: <sip:alias3@host3.example.com>
l: 0
```

3.1.1.6 Message with no LWS between display name and <

This OPTIONS request is not valid per the grammar in RFC 3261. since there is no LWS between the quoted string in the display name and < in the From header field value. This has been identified as a specification bug that will be removed when RFC 3261 is revised. Elements should accept this request as well formed.

Message Details : lwsdisp

```
OPTIONS sip:user@example.com SIP/2.0
To: sip:user@example.com
From: "caller"<sip:caller@example.com>;tag=323
Max-Forwards: 70
Call-ID: lwsdisp.1234abcd@funky.example.com
CSeq: 60 OPTIONS
Via: SIP/2.0/UDP funky.example.com;branch=z9hG4bKkdjuw
l: 0
```

3.1.1.7 Long values in header fields

This well-formed request contains header fields with many values and values that are very long. Features include:

- The To header field has a long display name, and long uri parameter names and values
- The From header field has long header parameter names and values, in particular a very long tag
- The Call-ID is one long token

Message Details : longreq

```
INVITE sip:user@example.com SIP/2.0
<allOneLine>
To: "I have a user name of
 <repeat count=10>extreme</repeat> proportion"
<sip:user@example.com:6000;
unknownparam1=very<repeat count=20>long</repeat>value;
longparam<repeat count=25>name</repeat>=shortvalue;
very<repeat count=25>long</repeat>ParameterNameWithNoValue>
</allOneLine>
<allOneLine>
F: sip:
<repeat count=5>amazinglylongcallername</repeat>@example.net
;tag=12<repeat count=50>982</repeat>424
;unknownheaderparam<repeat count=20>name</repeat>=
unknowheaderparam<repeat count=15>value</repeat>
;unknownValueless<repeat count=10>paramname</repeat>
</allOneLine>
Call-ID: longreq.one<repeat count=20>really</repeat>longcallid
CSeq: 3882340 INVITE
<allOneLine>
Unknown-<repeat count=20>Long</repeat>-Name:
 unknown-<repeat count=20>long</repeat>-value;
 unknown-<repeat count=20>long</repeat>-parameter-name =
```

```
 unknown-<repeat count=20>long</repeat>-parameter-value
</allOneLine>
Via: SIP/2.0/TCP sip33.example.com
v: SIP/2.0/TCP sip32.example.com
V: SIP/2.0/TCP sip31.example.com
Via: SIP/2.0/TCP sip30.example.com
ViA: SIP/2.0/TCP sip29.example.com
VIa: SIP/2.0/TCP sip28.example.com
VIA: SIP/2.0/TCP sip27.example.com
via: SIP/2.0/TCP sip26.example.com
viA: SIP/2.0/TCP sip25.example.com
vIa: SIP/2.0/TCP sip24.example.com
vIA: SIP/2.0/TCP sip23.example.com
V :  SIP/2.0/TCP sip22.example.com
v :  SIP/2.0/TCP sip21.example.com
V  : SIP/2.0/TCP sip20.example.com
v  : SIP/2.0/TCP sip19.example.com
Via : SIP/2.0/TCP sip18.example.com
Via  : SIP/2.0/TCP sip17.example.com
Via: SIP/2.0/TCP sip16.example.com
Via: SIP/2.0/TCP sip15.example.com
Via: SIP/2.0/TCP sip14.example.com
Via: SIP/2.0/TCP sip13.example.com
Via: SIP/2.0/TCP sip12.example.com
Via: SIP/2.0/TCP sip11.example.com
Via: SIP/2.0/TCP sip10.example.com
Via: SIP/2.0/TCP sip9.example.com
Via: SIP/2.0/TCP sip8.example.com
Via: SIP/2.0/TCP sip7.example.com
Via: SIP/2.0/TCP sip6.example.com
Via: SIP/2.0/TCP sip5.example.com
Via: SIP/2.0/TCP sip4.example.com
Via: SIP/2.0/TCP sip3.example.com
Via: SIP/2.0/TCP sip2.example.com
Via: SIP/2.0/TCP sip1.example.com
<allOneLine>
Via: SIP/2.0/TCP
 host.example.com;received=192.0.2.5;
branch=very<repeat count=50>long</repeat>branchvalue
</allOneLine>
Max-Forwards: 70
<allOneLine>
Contact: <sip:
<repeat count=5>amazinglylongcallername</repeat>
@host5.example.net>
</allOneLine>
Content-Type: application/sdp
l: 151
```

```
v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.1
s=-
c=IN IP4 192.0.2.1
t=0 0
m=audio 492170 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.1.8 Extra trailing octets in a UDP datagram

This message contains a single SIP REGISTER request, which ostensibly arrived over UDP in a single datagram. The packet contained extra octets after the body (which in this case has zero length). Those octets happen to look like a SIP INVITE request, but (per section 18.3 of [RFC3261]) they are just spurious noise that must be ignored.

A SIP element receiving this datagram would handle the REGISTER request normally and ignore the extra bits that look like an INVITE request. If the element is a proxy choosing to forward the REGISTER, the INVITE octets would not appear in the forwarded request.

Message Details : dblreq

```
REGISTER sip:example.com SIP/2.0
To: sip:j.user@example.com
From: sip:j.user@example.com;tag=43251j3j324
Max-Forwards: 8
I: dblreq.0ha0isndaksdj99sdfafnl3lk233412
Contact: sip:j.user@host.example.com
CSeq: 8 REGISTER
Via: SIP/2.0/UDP 192.0.2.125;branch=z9hG4bKkdjuw23492
Content-Length: 0


INVITE sip:joe@example.com SIP/2.0
t: sip:joe@example.com
From: sip:caller@example.net;tag=141334
Max-Forwards: 8
Call-ID: dblreq.0ha0isnda977644900765@192.0.2.15
CSeq: 8 INVITE
Via: SIP/2.0/UDP 192.0.2.15;branch=z9hG4bKkdjuw380234
Content-Type: application/sdp
Content-Length: 151

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.15
s=-
c=IN IP4 192.0.2.15
t=0 0
m=audio 492170 RTP/AVP 0 12
m =video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.1.9 Semicolon separated parameters in URI user part

This request has a semicolon-separated parameter contained in the "user" part of the Request-URI (whose value contains an escaped @ symbol). Receiving elements will accept this as a well formed message. The Request-URI will parse such that the user part is "user;par=u@example.net".

Message Details : semiuri

```
OPTIONS sip:user;par=u%40example.net@example.com SIP/2.0
To: sip:j_user@example.com
From: sip:caller@example.org;tag=33242
Max-Forwards: 3
Call-ID: semiuri.0ha0isndaksdj
CSeq: 8 OPTIONS
Accept: application/sdp, application/pkcs7-mime,
        multipart/mixed, multipart/signed,
        message/sip, message/sipfrag
Via: SIP/2.0/UDP 192.0.2.1;branch=z9hG4bKkdjuw
l: 0
```

3.1.1.10 Varied and unknown transport types

This request contains Via header field values with all known transport types and exercises the transport extension mechanism. Parsers must accept this message as well formed. Elements receiving this message would process it exactly as if the 2nd and subsequent header field values specified UDP (or other transport).

Message Details : transports

```
OPTIONS sip:user@example.com SIP/2.0
To: sip:user@example.com
From: <sip:caller@example.com>;tag=323
Max-Forwards: 70
Call-ID:  transports.kijh4akdnaqjkwendsasfdj
Accept: application/sdp
CSeq: 60 OPTIONS
Via: SIP/2.0/UDP t1.example.com;branch=z9hG4bKkdjuw
Via: SIP/2.0/SCTP t2.example.com;branch=z9hG4bKklasjdhf
Via: SIP/2.0/TLS t3.example.com;branch=z9hG4bK2980unddj
Via: SIP/2.0/UNKNOWN t4.example.com;branch=z9hG4bKasd0f3en
Via: SIP/2.0/TCP t5.example.com;branch=z9hG4bK0a9idfnee
l: 0
```

3.1.1.11 S/MIME signed message

This is a signed INVITE request. The signature is binary encoded. The body contains null (0x00) characters. Receivers must take care to properly frame the received message.

Parsers must accept this message as well formed, even if the application above the parser does not support multipart/signed.

Message Details : smime01

```
INVITE sip:receiver@example.com SIP/2.0
Via: SIP/2.0/UDP host5.example.org;branch=z9hG4bK923rnasdkl3
To: <sip:receiver@example.com>
From: <sip:sender@example.org>;tag=2390234seiu3
Call-ID: smime01.uoqeiuavnklafekjq34iu43uawe
CSeq: 282398492 INVITE
Max-Forwards: 70
Contact: <sip:sender@host5.example.org>
Content-Length: 3134
Content-Type: multipart/signed;
              protocol="application/pkcs-7-signature";
              micalg=sha1;
              boundary="----EABF38A0AAE8704C560F10418BA807CF"

------EABF38A0AAE8704C560F10418BA807CF
Content-Type: message/sip

INVITE sip:receiver@example.com SIP/2.0
Via: SIP/2.0/UDP host5.example.org;branch=z9hG4bK923rnasdkl3
To: <sip:receiver@example.com>
From: <sip:sender@example.org>;tag=2390234seiu3
Call-ID: smime01.uoqeiuavnklafekjq34iu43uawe
CSeq: 282398492 INVITE
Max-Forwards: 70
Contact: <sip:sender@host5.example.org>
Accept: application/sdp, application/pkcs7-mime,
        multipart/mixed, multipart/signed,
        message/sip, message/sipfrag
Content-Type: application/sdp
Content-Length: 149

v=0
o=sender 29739 7272939 IN IP4 192.0.2.1
s=-
c=IN IP4 192.0.2.1
t=0 0
m=audio 492170 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC

------EABF38A0AAE8704C560F10418BA807CF
Content-Type: application/pkcs-7-signature; name="smime.p7s"
Content-Transfer-Encoding: binary
```

Content-Disposition: attachment; filename="smime.p7s"

<hex>3082088806092A86
4886F70D010702A0820879308208750201013 10B300906052B0E03021A050030
0B06092A864886F70D010701A082067A30820339308202A2A003020102020800
900089022400013 00D06092A864886F70D0101050500307 0310B300906035504
0613025553311330110603550408130A43616C69666F726E69613111300F0603
550407130853616E4A6F7365310E300C060355040A13057369706974312930 27
060355040B13536970697454657374436572746966696361746541757 4686F72
697479301E170D3033313032313134343332355A170D3133313031383134343 3
32355A3062310B3009060355040613025553311330110603550408130A43616C
69666F726E69613111300F060355040713085361 6E4A6F7365310E300C060355
040A13057369706974311B3019060355040314127365 6E646572406578616D70
6C652E6F726730819F300D06092A864886F70D010101050003818D0030818902
818100CB8302060F12C8FA2D1786922CA173DCEB80BF1B1B8AF74A310C6975A5
56A7630FB6E044D9E994DCD49AFF7976C462D7A8E74ECBF98723AEBF2796EDDD
6263577C6C2B77DC7C300B533DEDB5FB8EB3827FD6FC9B37B9A0DE829F1B1081
D632A8AD9FB00A860928E88F87E0B979BA65294AC7D6D2D18A78C86B4FA73387
4E230203010001A381E93081E6301D0603551D110416301481127 3656E646572
406578616D706C652E6F726730090603551D1304023000301D0603551D0E0416
041440FF1C0C1BB8684CA917839D70E97DF8DD5B60D130819A0603551D230481
9230818F80146B461714EA94762580546E1354DAA1E35414A1B6A174A4723070
310B3009060355040613025553311330110603550408130A43616C69666F726E
69613111300F060355040713085361 6E4A6F7365310E300C060355040A130573
697069743129302706035504 0B13536970697454657374436572746966696361
7465417574686F726974782010030 0D06092A864886F70D0101050500038181
006FFE1A3B5CE807C3DD2CFDF6E9787F491C84DBF7DCD11DB2D6A8887D2FE3F2
2E9C6894994282E50AA0DFFE1CBD4EC2C20217831FC2AD360FF1C0DE1DE1E870
102CFA99EE504C7DC0D8752A63294AC748DDDEFADE55C6D051F1CD54CFE7C153
278962A53CEF61B875C1FD3C74E972242CBA0131B3B8C607BF95B378212CA9A7
5E30820339308202A2A003020102020800900089022400013 00D06092A864886
F70D0101050500307 0310B300906035504061302555331133011060355040813
0A43616C69666F726E69613111300F060355040713085361 6E4A6F7365310E30
0C060355040A13057369706974312930 27060355040B13536970697454657374
436572746966696361746541757 4686F72697479301E170D3033313032313134
343332355A170D3133313031383134343332355A3062310B3009060355040613
025553311330110603550408130A43616C69666F726E69613111300F06035504
07130853616E4A6F7365310E300C060355040A13057369706974311B30190603
550403141273656E646572406578616D706C652E6F726730819F300D06092A86
4886F70D0101010500038 18D0030818902818100CB8302060F12C8FA2D178692
2CA173DCEB80BF1B1B8AF74A310C6975A556A7630FB6E044D9E994DCD49AFF79
76C462D7A8E74ECBF98723AEBF2796EDDD6263577C6C2B77DC7C300B533DEDB5
FB8EB3827FD6FC9B37B9A0DE829F1B1081D632A8AD9FB00A860928E88F87E0B9
79BA65294AC7D6D2D18A78C86B4FA733874E230203010001A381E93081E6301D
0603551D110416301481127 3656E646572406578616D706C652E6F726730090 6
03551D1304023000301D0603551D0E0416041440FF1C0C1BB8684CA917839D70
E97DF8DD5B60D130819A0603551D2304819230818F80146B461714EA94762580
546E1354DAA1E35414A1B6A174A4723070310B3009060355040613025553311 3

```
30110603550408130A43616C69666F726E69613111300F060355040713085361
6E4A6F7365310E300C060355040A13057369706974312930270603550408B1353
697069745465737443657274696669636174654175746F7269747982010030
0D06092A864886F70D0101050500038181006FFE1A3B5CE807C3DD2CFDF6E978
7F491C84DBF7DCD11DB2D6A8887D2FE3F22E9C6894994282E50AA0DFFE1CBD4E
C2C20217831FC2AD360FF1C0DE1DE1E870102CFA99EE504C7DC0D8752A63294A
C748DDDEFADE55C6D051F1CD54CFE7C153278962A53CEF61B875C1FD3C74E972
242CBA0131B3B8C607BF95B378212CA9A75E318201D6308201D2020101307C30
70310B300906035504061302555331133011060355040813 0A43616C69666F72
6E69613111300F060355040713 0853616E4A6F7365310E300C060355040A1305
736970697431293027060355040B13536970697454657374436572746966696 3
617465417574686F726974790208009000890224000130090605 2B0E03021A05
00A081B1301806092A864886F70D010903310B06092A864886F70D010701301C
06092A864886F70D010905310F170D3034303731343230333630375A30230609
2A864886F70D01090431160414 2830DBEDC758BAA9750138384431E8F1819717
C8305206092A864886F70D01090F31453043300A06082A864886F70D0307300E
06082A864886F70D03020202008030 0D06082A864886F70D0302020140300706
052B0E030207300D06082A864886F70D0302020128300D06092A864886F70D01
0101050004818060FF00161CFAC1E641D0C993FACD949DC239F9DDF527F834A6
ECF837B5D2C7B756F30A2755F5EF7A833351721F30A4392921E0A73594AA0BF2
4E6BD3981362DEA4F22844CB505ED3C6C0CCADD524066593ABED4023243055CA
0CE0AE83F13868E29E91E3AA588BDC11B2A18F39001885E17879503523699A87
982C6D48D4918A</hex>------EABF38A0AAE8704C560F10418BA807CF--
```

3.1.1.12 Unusual reason phrase

This 200 response contains a reason phrase other than "OK". The reason phrase is intended for human consumption, and may contain any string produced by

```
Reason-Phrase   =  *(reserved / unreserved / escaped
                   / UTF8-NONASCII / UTF8-CONT / SP / HTAB)
```

This particular response contains unreserved and non-ASCII UTF-8 characters.This response is well formed. A parser must accept this message.

```
Message Details : unreason

<allOneLine>
SIP/2.0 200 = 2**3 * 5**2 <hex>D0BDD0BE20D181D182
D0BE20D0B4D0B5D0B2D18FD0BDD0BED181D182D0BE20D0B4
D0B5D0B2D18FD182D18C202D20D0BFD180D0BED181D182D0
BED0B5</hex>
</allOneLine>
Via: SIP/2.0/UDP 192.0.2.198;branch=z9hG4bK1324923
Call-ID: unreason.1234ksdfak3j2erwedfsASdf
CSeq: 35 INVITE
From: sip:user@example.com;tag=11141343
To: sip:user@example.edu;tag=2229
Content-Length: 159
Content-Type: application/sdp
Contact: <sip:user@host198.example.com>

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.198
s=-
c=IN IP4 192.0.2.198/127
t=0 0
m=audio 492170 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.1.13 Empty reason phrase

This well formed response contains no reason phrase. A parser must accept this message. The space character after the reason code is required. If it were not present, this message could be rejected as invalid (a liberal receiver would accept it anyway).

```
Message Details : noreason

SIP/2.0 100<hex>20</hex>
Via: SIP/2.0/UDP 192.0.2.105;branch=z9hG4bK2398ndaoe
Call-ID: noreason.asndj203insdf99223ndf
CSeq: 35 INVITE
From: <sip:user@example.com>;tag=39ansfi3
To: <sip:user@example.edu>;tag=902jndnke3
Content-Length: 0
Contact: <sip:user@host105.example.com>
```

3.1.2 Invalid messages

This section contains several invalid messages reflecting errors seen at interoperability events and exploring important edge conditions that can be induced through malformed messages. This section does not attempt to be a comprehensive list of all types of invalid messages.

3.1.2.1 Extraneous header field separators

The Via and header field of this request contains contain additional semicolons and commas without parameters or values. The Contact header field contains additional semicolons without parameters. This message is syntactically invalid.

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

```
Message Details : badinv01

INVITE sip:user@example.com SIP/2.0
To: sip:j.user@example.com
From: sip:caller@example.net;tag=134161461246
Max-Forwards: 7
Call-ID: badinv01.0ha0isndaksdjasdf3234nas
CSeq: 8 INVITE
Via: SIP/2.0/UDP 192.0.2.15;;,;,,
Contact: "Joe" <sip:joe@example.org>;;;;
Content-Length: 153
Content-Type: application/sdp

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.15
s=-
c=IN IP4 192.0.2.15
t=0 0
m=audio 492170 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.2.2 Content length larger than message

This is a request message with a Content Length that is larger than the length of the body.

When sent UDP (as this message ostensibly was), the receiving element should respond with a 400 Bad Request error. If this message were

received over a stream-based transport such as TCP, there's not much you can do but wait for more data on the stream and close the connection if none is forthcoming in a reasonable period of time.

Message Details : clerr

```
INVITE sip:user@example.com SIP/2.0
Max-Forwards: 80
To: sip:j.user@example.com
From: sip:caller@example.net;tag=9342939o2
Contact: <sip:caller@hungry.example.net>
Call-ID: clerr.0ha0isndaksdjweiafasdk3
CSeq: 8 INVITE
Via: SIP/2.0/UDP host5.example.com;branch=z9hG4bK-39234-23523
Content-Type: application/sdp
Content-Length: 9999

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.155
s=-
c=IN IP4 192.0.2.155
t=0 0
m=audio 492170 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.2.3 Negative Content-Length

This request has a negative value for Content-Length.

An element receiving this message should respond with an error. This request appeared over UDP, so the remainder of the datagram can simply be discarded. If a request like this arrives over TCP, the framing error is not recoverable and the connection should be closed. The same behavior is appropriate for messages that arrive without a numeric value in the Content-Length header field such as:

```
Content-Length: five
```

Implementors should take extra precautions if the technique they choose for converting this ascii field into an integral form can return a negative value. In particular, the result must not be used as a counter or array index.

```
Message Details : ncl

INVITE sip:user@example.com SIP/2.0
Max-Forwards: 254
To: sip:j.user@example.com
From: sip:caller@example.net;tag=32394234
Call-ID: ncl.0ha0isndaksdj2193423r542w35
CSeq: 0 INVITE
Via: SIP/2.0/UDP 192.0.2.53;branch=z9hG4bKkdjuw
Contact: <sip:caller@example53.example.net>
Content-Type: application/sdp
Content-Length: -999

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.53
s=-
c=IN IP4 192.0.2.53
t=0 0
m=audio 492170 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.2.4 Request scalar fields with overlarge values

This request contains several scalar header field values outside their legal range.

- the CSeq sequence number is >2**32-1.
- the Max-Forwards value is >255.
- the Expires value is >2**32-1.
- the Contact expires parameter value is >2**32-1.

An element receiving this request should respond with a 400 Bad Request due to the CSeq error. If only the Max-Forwards field were in error, the element could choose process the request as if the field were absent. If only the expiry values were in error, the element could treat them as if they contained the default values for expiration (3600 in this case).

Other scalar request fields that may contain aberrant values include, but are not limited to, the Contact q value, the Timestamp value, and the Via ttl parameter.

```
Message Details : scalar02

REGISTER sip:example.com SIP/2.0
Via: SIP/2.0/TCP host129.example.com;branch=z9hG4bK342sdfoi3
To: <sip:user@example.com>
From: <sip:user@example.com>;tag=239232jh3
CSeq: 36893488147419103232 REGISTER
Call-ID: scalar02.23o0pd9vanlq3wnrlnewofjas9ui32
Max-Forwards: 300
Expires: 1<repeat count=100>0</repeat>
Contact: <sip:user@host129.example.com>
  ;expires=280297596632815
Content-Length: 0
```

3.1.2.5 Response scalar fields with overlarge values

This response contains several scalar header field values outside their legal range.

- the CSeq sequence number is >2**32-1.
- The Retry-After field is unreasonably large (note that RFC 3261 does not define a legal range for this field).
- The Warning field has a warning-value with more than 3 digits

An element receiving this response will simply discard it.

```
Message Details : scalarlg

SIP/2.0 503 Service Unavailable
Via: SIP/2.0/TCP host129.example.com;branch=z0hG4bKzzxdiwo34sw
To: <sip:user@example.com>
From: <sip:other@example.net>;tag=2easdjfejw
CSeq: 9292394834772304023312 OPTIONS
Call-ID: scalarlg.noase0of0234hn2qofoaf0232aewf2394r
Retry-After: 94930283850302834930402398 8
Warning: 1812 overture "In Progress"
Content-Length: 0
```

3.1.2.6 Unterminated quoted string in display-name

This is a request with an unterminated quote in the display name of the To field. An element receiving this request should return an 400 Bad Request error.

An element could attempt to infer a terminating quote and accept the

message. Such an element needs to take care that it makes a reasonable inference when it encounters

To: "Mr J. User <sip:j.user@example.com> <sip:realj@example.net>

Message Details : quotbal

```
INVITE sip:user@example.com SIP/2.0
To: "Mr. J. User <sip:j.user@example.com>
From: sip:caller@example.net;tag=93334
Max-Forwards: 10
Call-ID: quotbal.aksdj
Contact: <sip:caller@host59.example.net>
CSeq: 8 INVITE
Via: SIP/2.0/UDP 192.0.2.59:5050;branch=z9hG4bKkdjuw39234
Content-Type: application/sdp
Content-Length: 153

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.15
s=-
c=IN IP4 192.0.2.15
t=0 0
m=audio 492170 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.2.7 <> enclosing Request-URI

This INVITE request is invalid because the Request-URI has been enclosed within in "<>".

It is reasonable to always reject a request with this error with a 400 Bad Request. Elements attempting to be liberal with what they accept may choose to ignore the brackets. If the element forwards the request, it must not include the brackets in the messages it sends.

Message Details : ltgtruri

```
INVITE <sip:user@example.com> SIP/2.0
To: sip:user@example.com
From: sip:caller@example.net;tag=39291
Max-Forwards: 23
Call-ID: ltgtruri.1@192.0.2.5
CSeq: 1 INVITE
Via: SIP/2.0/UDP 192.0.2.5
Contact: <sip:caller@host5.example.net>
Content-Type: application/sdp
Content-Length: 160

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.5
s=-
c=IN IP4 192.0.2.5
t=3149328700 0
m=audio 492170 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.2.8 Malformed SIP Request-URI (embedded LWS)

This INVITE has illegal LWS within the Request-URI.

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

An element could attempt to ignore the embedded LWS for those schemes (like sip) where that would not introduce ambiguity.

```
Message Details : lwsruri

INVITE sip:user@example.com; lr SIP/2.0
To: sip:user@example.com;tag=3xfe-9921883-z9f
From: sip:caller@example.net;tag=231413434
Max-Forwards: 5
Call-ID: lwsruri.asdfasdoeoi2323-asdfwrn23-asd834rk423
CSeq: 2130706432 INVITE
Via: SIP/2.0/UDP 192.0.2.1:5060;branch=z9hG4bKkdjuw2395
Contact: <sip:caller@host1.example.net>
Content-Type: application/sdp
Content-Length: 160

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.1
s=-
c=IN IP4 192.0.2.1
t=3149328700 0
m=audio 492170 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.2.9 Multiple SP separating Request-Line elements

This INVITE has illegal multiple SP characters between elements of the start line.

It is acceptable to reject this request as malformed. An element that is liberal in what it accepts may ignore these extra SP characters while processing the request. If the element forwards the request, it must not include these extra SP characters in the messages it sends.

Message Details : lwsstart

```
INVITE  sip:user@example.com  SIP/2.0
Max-Forwards: 8
To: sip:user@example.com
From: sip:caller@example.net;tag=8814
Call-ID: lwsstart.dfknq234oi243099adsdfnawe3@example.com
CSeq: 1893884 INVITE
Via: SIP/2.0/UDP host1.example.com;branch=z9hG4bKkdjuw3923
Contact: <sip:caller@host1.example.net>
Content-Type: application/sdp
Content-Length: 151

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.1
s=-
c=IN IP4 192.0.2.1
t=0 0
m=audio 492170 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.2.10 SP characters at end of Request-Line

This OPTIONS request contains SP characters between the SIP-Version field and the CRLF terminating the Request-Line.

It is acceptable to reject this request as malformed. An element that is liberal in what it accepts may ignore these extra SP characters while processing the request. If the element forwards the request, it must not include these extra SP characters in the messages it sends.

Message Details : trws

```
OPTIONS sip:remote-target@example.com SIP/2.0<hex>2020</hex>
Via: SIP/2.0/TCP host1.examle.com;branch=z9hG4bK299342093
To: <sip:remote-target@example.com>
From: <sip:local-resource@example.com>;tag=329429089
Call-ID: trws.oicu34958239neffasdhr2345r
Accept: application/sdp
CSeq: 238923 OPTIONS
Max-Forwards: 70
Content-Length: 0
```

3.1.2.11 Escaped headers in SIP Request-URI

This INVITE is malformed as the SIP Request-URI contains escaped headers.

It is acceptable for an element to reject this request with a 400 Bad Request. An element could choose to be liberal in what it accepts and ignore the escaped headers. If the element is a proxy, the escaped headers must not appear in the Request-URI of forwarded request (and most certainly must not be translated into the actual header of the forwarded request).

```
Message Details : escruri

INVITE sip:user@example.com?Route=%3Csip:example.com%3E SIP/2.0
To: sip:user@example.com
From: sip:caller@example.net;tag=341518
Max-Forwards: 7
Contact: <sip:caller@host39923.example.net>
Call-ID: escruri.23940-asdfhj-aje3br-234q098w-fawerh2q-h4n5
CSeq: 149209342 INVITE
Via: SIP/2.0/UDP host-of-the-hour.example.com;branch=z9hG4bKkdjuw
Content-Type: application/sdp
Content-Length: 151

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.1
s=-
c=IN IP4 192.0.2.1
t=0 0
m=audio 492170 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.2.12 Invalid timezone in Date header field

This INVITE is invalid as it contains a non GMT time zone in the SIP Date header field.

It is acceptable to reject this request as malformed (though an element shouldn't do that unless the contents of the Date header field were actually important to its processing). An element wishing to be liberal in what it accepts could ignore this value altogether if it wasn't going to use the Date header field anyhow. Otherwise, it could attempt to interpret this date and adjust it to GMT.

RFC 3261 explicitly defines the only acceptable timezone designation

as "GMT". "UT", while synonymous with GMT per [RFC2822], is not valid. "UTC" and "UCT" are also invalid.

Message Details : baddate

```
INVITE sip:user@example.com SIP/2.0
To: sip:user@example.com
From: sip:caller@example.net;tag=2234923
Max-Forwards: 70
Call-ID: baddate.239423mnsadf3j23lj42--sedfnm234
CSeq: 1392934 INVITE
Via: SIP/2.0/UDP host.example.com;branch=z9hG4bKkdjuw
Date: Fri, 01 Jan 2010 16:00:00 EST
Contact: <sip:caller@host5.example.net>
Content-Type: application/sdp
Content-Length: 151

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.5
s=-
c=IN IP4 192.0.2.5
t=0 0
m=audio 492170 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.2.13 Failure to enclose name-addr URI in <>

This REGISTER request is malformed. The SIP URI contained in the Contact Header field has an escaped header, so the field must be in name-addr form (which implies the URI must be enclosed in <>).

It is reasonable for an element receiving this request to respond with a 400 Bad Request. An element choosing to be liberal in what it accepts could infer the angle brackets since there is no ambiguity in this example. In general, that won't be possible.

Message Details : regbadct

```
REGISTER sip:example.com SIP/2.0
To: sip:user@example.com
From: sip:user@example.com;tag=998332
Max-Forwards: 70
Call-ID: regbadct.k345asrl3fdbv@10.0.0.1
CSeq: 1 REGISTER
Via: SIP/2.0/UDP 135.180.130.133:5060;branch=z9hG4bKkdjuw
Contact: sip:user@example.com?Route=%3Csip:sip.example.com%3E
```

```
l: 0
```

3.1.2.14 Spaces within addr-spec

This request is malformed since the addr-spec in the To header field contains spaces. Parsers receiving this request must not break. It is reasonable to reject this request with a 400 Bad Request response. Elements attempting to be liberal may ignore the spaces.

Message Details : badaspec

```
OPTIONS sip:user@example.org SIP/2.0
Via: SIP/2.0/UDP host4.example.com:5060;branch=z9hG4bKkdju43234
Max-Forwards: 70
From: "Bell, Alexander" <sip:a.g.bell@example.com>;tag=433423
To: "Watson, Thomas" < sip:t.watson@example.org >
Call-ID: badaspec.sdf0234n2nds0a099u23h3hnnw009cdkne3
Accept: application/sdp
CSeq: 3923239 OPTIONS
l: 0
```

3.1.2.15 Non-token characters in display-name

This OPTIONS request is malformed since the display names in the To and From header fields contain non-token characters but are unquoted.

It is reasonable to always reject this kind of error with a 400 Bad Request response.

An element may attempt to be liberal in what it receives and infer the missing quotes. If this element were a proxy, it must not propagate the error into the request it forwards. As a consequence, if the fields are covered by a signature, there's not much point in trying to be liberal - the message should be simply rejected.

Message Details : baddn

```
OPTIONS sip:t.watson@example.org SIP/2.0
Via:     SIP/2.0/UDP c.example.com:5060;branch=z9hG4bKkdjuw
Max-Forwards:      70
From:    Bell, Alexander <sip:a.g.bell@example.com>;tag=43
To:      Watson, Thomas <sip:t.watson@example.org>
Call-ID: baddn.31415@c.example.com
Accept: application/sdp
```

```
CSeq:    3923239 OPTIONS
l: 0
```

3.1.2.16 Unknown protocol version

To an element implementing [RFC3261], this request is malformed due to its high version number.

The element should respond to the request with a 505 Version Not Supported error.

```
Message Details : badvers

OPTIONS sip:t.watson@example.org SIP/7.0
Via:     SIP/7.0/UDP c.example.com;branch=z9hG4bKkdjuw
Max-Forwards:     70
From:    A. Bell <sip:a.g.bell@example.com>;tag=qweoiqpe
To:      T. Watson <sip:t.watson@example.org>
Call-ID: badvers.31417@c.example.com
CSeq:    1 OPTIONS
l: 0
```

3.1.2.17 Start line and CSeq method mismatch

This request has mismatching values for the method in the start line and the CSeq header field. Any element receiving this request will respond with a 400 Bad Request.

```
Message Details : mismatch01

OPTIONS sip:user@example.com SIP/2.0
To: sip:j.user@example.com
From: sip:caller@example.net;tag=34525
Max-Forwards: 6
Call-ID: mismatch01.dj0234sxdfl3
CSeq: 8 INVITE
Via: SIP/2.0/UDP host.example.com;branch=z9hG4bKkdjuw
l: 0
```

3.1.2.18 Unknown Method with CSeq method mismatch

This message has an unknown method, and a CSeq method tag which does not match it.

Any element receiving this response will should respond with a 501 Not Implemented. A 400 Bad Request is also acceptable, but choosing a 501 (particularly at proxies) has better future-proof characteristics.

Message Details : mismatch02

```
NEWMETHOD sip:user@example.com SIP/2.0
To: sip:j.user@example.com
From: sip:caller@example.net;tag=34525
Max-Forwards: 6
Call-ID: mismatch02.dj0234sxdfl3
CSeq: 8 INVITE
Contact: <sip:caller@host.example.net>
Via: SIP/2.0/UDP host.example.net;branch=z9hG4bKkdjuw
Content-Type: application/sdp
l: 139

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.1
c=IN IP4 192.0.2.1
m=audio 492170 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.2.19 Overlarge response code

This response has a response code larger than 699. An element receiving this response should simply drop it.

Message Details : bigcode

```
SIP/2.0 4294967301 better not break the receiver
Via: SIP/2.0/UDP 192.0.2.105;branch=z9hG4bK2398ndaoe
Call-ID: bigcode.asdof3uj203asdnf3429uasdhfas3ehjasdfas9i
CSeq: 353494 INVITE
From: <sip:user@example.com>;tag=39ansfi3
To: <sip:user@example.edu>;tag=902jndnke3
Content-Length: 0
Contact: <sip:user@host105.example.com>
```

3.2 Transaction layer semantics

This section contains tests that exercise an implementation's parser and transaction layer logic.

3.2.1 Missing transaction identifier

This request indicates support for RFC 3261-style transaction identifiers by providing the z9hG4bK prefix to the branch parameter, but it provides no identifier. A parser must not break when receiving this message. An element receiving this request could reject the request with a 400 Response (preferably statelessly, as other requests from the source are likely to also have a malformed branch parameter), or it could fall back to the RFC 2543 style transaction identifier.

```
Message Details : badbranch

OPTIONS sip:user@example.com SIP/2.0
To: sip:user@example.com
From: sip:caller@example.org;tag=33242
Max-Forwards: 3
Via: SIP/2.0/UDP 192.0.2.1;branch=z9hG4bK
Accept: application/sdp
Call-ID: badbranch.sadonfo23i420jv0as0derf3j3n
CSeq: 8 OPTIONS
l: 0
```

3.3 Application layer semantics

This section contains tests that exercise an implementation's parser and application layer logic.

3.3.1 Missing Required Header Fields

This request contains no Call-ID, From, or To header fields.

An element receiving this message must not break because of the missing information. Ideally, it will respond with a 400 Bad Request error.

```
Message Details : insuf

INVITE sip:user@example.com SIP/2.0
CSeq: 193942 INVITE
Via: SIP/2.0/UDP 192.0.2.95;branch=z9hG4bKkdj.insuf
Content-Type: application/sdp
l: 153

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.95
s=-
c=IN IP4 192.0.2.95
t=0 0
m=audio 492170 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.3.2 Request-URI with unknown scheme

This OPTIONS contains an unknown URI scheme in the Request-URI. A parser must accept this as a well-formed SIP request.

An element receiving this request will reject it with a 416 Unsupported URI Scheme response.

Some early implementations attempt to look at the contents of the To header field to determine how to route this kind of request. That is an error. Despite the fact that the To header field and the Request URI frequently look alike in simplistic first-hop messages, the To header field contains no routing information.

```
Message Details : unkscm

OPTIONS nobodyKnowsThisScheme:totallyopaquecontent SIP/2.0
To: sip:user@example.com
From: sip:caller@example.net;tag=384
Max-Forwards: 3
Call-ID: unkscm.nasdfasser0q239nwsdfasdkl34
CSeq: 3923423 OPTIONS
Via: SIP/2.0/TCP host9.example.com;branch=z9hG4bKkdjuw39234
Content-Length: 0
```

3.3.3 Request-URI with known but atypical scheme

This OPTIONS contains an Request-URI with an IANA registered scheme

that does not commonly appear Request-URIs of SIP requests. A parser must accept this as a well-formed SIP request.

If an element will never accept this scheme as meaningful in a request-URI, it is appropriate to treat it as unknown and return a 416 Unsupported URI Scheme response. If the element might accept some URIs with this scheme, then a 404 Not Found is appropriate for those URIs it doesn't accept.

Message Details : novelsc

```
OPTIONS soap.beep://192.0.2.103:3002 SIP/2.0
To: sip:user@example.com
From: sip:caller@example.net;tag=384
Max-Forwards: 3
Call-ID: novelsc.asdfasser0q239nwsdfasdkl34
CSeq: 3923423 OPTIONS
Via: SIP/2.0/TCP host9.example.com;branch=z9hG4bKkdjuw39234
Content-Length: 0
```

3.3.4 Unknown URI schemes in header fields

This message contains registered schemes in the To, From and Contact header fields of a request. The message is syntactically valid. Parsers must not fail when receiving this message.

Proxies should treat this message as they would any other request for this URI. A registrar would reject this request with a 400 Bad Request response since the To: header field is required to contain a SIP or SIPS URI as an AOR.

Message Details : unksm2

```
REGISTER sip:example.com SIP/2.0
To: isbn:2983792873
From: <http://www.example.com>;tag=3234233
Call-ID: unksm2.daksdj@hyphenated-host.example.com
CSeq: 234902 REGISTER
Max-Forwards: 70
Via: SIP/2.0/UDP 192.0.2.21:5060;branch=z9hG4bKkdjuw
Contact: <name:John_Smith>
l: 0
```

3.3.5 Proxy-Require and Require

This request tests proper implementation of SIP's Proxy-Require and Require extension mechanisms.

Any element receiving this request will respond with a 420 Bad Extension response containing an Unsupported header field listing these features from either the Require or Proxy-Require header field depending on the role in which the element is responding.

Message Details : bext01

```
OPTIONS sip:user@example.com SIP/2.0
To: sip:j_user@example.com
From: sip:caller@example.net;tag=242etr
Max-Forwards: 6
Call-ID: bext01.0ha0isndaksdj
Require: nothingSupportsThis, nothingSupportsThisEither
Proxy-Require: noProxiesSupportThis, norDoAnyProxiesSupportThis
CSeq: 8 OPTIONS
Via: SIP/2.0/TLS fold-and-staple.example.com;branch=z9hG4bKkdjuw
Content-Length: 0
```

3.3.6 Unknown Content-Type

This INVITE request contains a body of unknown type. It is syntactically valid. A parser must not fail when receiving it.

A proxy receiving this request would process it just like any other INVITE. An endpoint receiving this request would reject it with a 415 Unsupported Media Type error.

Message Details : invut

```
INVITE sip:user@example.com SIP/2.0
Contact: <sip:caller@host5.example.net>
To: sip:j.user@example.com
From: sip:caller@example.net;tag=8392034
Max-Forwards: 70
Call-ID: invut.0ha0isndaksdjadsfij34n23d
CSeq: 235448 INVITE
Via: SIP/2.0/UDP somehost.example.com;branch=z9hG4bKkdjuw
Content-Type: application/unknownformat
Content-Length: 40

<audio>
```

```
      <pcmu port="443"/>
    </audio>
```

3.3.7 Unknown authorization scheme

This REGISTER request contains an Authorization header field with an unknown scheme. The request is well-formed. A parser must not fail when receiving it.

A proxy will treat this request as any other REGISTER. If it forwards the request, it will include this Authorization header field unmodified in the forwarded messages.

A registrar that does not care about challenge-response authentication will simply ignore the Authorization header field, processing this registration as if the field were not present. A registrar that does care about challenge-response authentication will reject this request with a 401, issuing a new challenge with a scheme it understands.

Endpoints choosing not to act as registrars will simply reject the request. A 405 Method Not Allowed is appropriate.

```
Message Details : regaut01

REGISTER sip:example.com SIP/2.0
To: sip:j.user@example.com
From: sip:j.user@example.com;tag=87321hj23128
Max-Forwards: 8
Call-ID: regaut01.0ha0isndaksdj
CSeq: 9338 REGISTER
Via: SIP/2.0/TCP 192.0.2.253;branch=z9hG4bKkdjuw
Authorization: NoOneKnowsThisScheme opaque-data=here
Content-Length:0
```

3.3.8 Multiple values in single value required fields

The message contains a request with multiple Call-ID, To, From, Max-Forwards and CSeq values. An element receiving this request must not break.

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

```
Message Details : multi01

INVITE sip:user@company.com SIP/2.0
Contact: <sip:caller@host25.example.net>
Via: SIP/2.0/UDP 192.0.2.25;branch=z9hG4bKkdjuw
Max-Forwards: 70
CSeq: 5 INVITE
Call-ID: multi01.98asdh@192.0.2.1
CSeq: 59 INVITE
Call-ID: multi01.98asdh@192.0.2.2
From: sip:caller@example.com;tag=3413415
To: sip:user@example.com
To: sip:other@example.net
From: sip:caller@example.net;tag=2923420123
Content-Type: application/sdp
l: 155
Contact: <sip:caller@host36.example.net>
Max-Forwards: 5

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.25
s=-
c=IN IP4 192.0.2.25
t=0 0
m=audio 492170 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.3.9 Multiple Content-Length values

Multiple conflicting Content-Length header field values appear in this request.

From a framing perspective, this situation is equivalent to an invalid Content-Length value (or no value at all).

An element receiving this message should respond with an error. This request appeared over UDP, so the remainder of the datagram can simply be discarded. If a request like this arrives over TCP, the framing error is not recoverable and the connection should be closed.

Message Details : mcl01

```
OPTIONS sip:user@example.com SIP/2.0
Via: SIP/2.0/UDP host5.example.net;branch=z9hG4bK293423
To: sip:user@example.com
From: sip:other@example.net;tag=3923942
Call-ID: mcl01.fhn2323orihawfdoa3o4r52o3irsdf
CSeq: 15932 OPTIONS
Content-Length: 13
Max-Forwards: 60
Content-Length: 5
Content-Type: text/plain

There's no way to know how many octets are supposed to be here.
```

3.3.10 200 OK Response with broadcast Via header field value

This message is a response with a 2nd Via header field value's sent-by containing 255.255.255.255. The message is well formed - parsers must not fail when receiving it.

Per [RFC3261] an endpoint receiving this message should simply discard it.

If a proxy followed normal response processing rules blindly, it would forward this response to the broadcast address. To protect against this being used as an avenue of attack, proxies should drop such responses.

```
Message Details : bcast

SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.0.2.198;branch=z9hG4bK1324923
Via: SIP/2.0/UDP 255.255.255.255;branch=z9hG4bK1saber23
Call-ID: bcast.0384840201234ksdfak3j2erwedfsASdf
CSeq: 35 INVITE
From: sip:user@example.com;tag=11141343
To: sip:user@example.edu;tag=2229
Content-Length: 159
Content-Type: application/sdp
Contact: <sip:user@host28.example.com>

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.198
s=-
c=IN IP4 192.0.2.198/127
t=0 0
m=audio 492170 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.3.11 Max-Forwards of zero

This is a legal SIP request with the Max-Forwards header field value set to zero.

A proxy should not forward the request and respond 483 (Too Many Hops). An endpoint should process the request as if the Max-Forwards field value were still positive.

```
Message Details : zeromf

OPTIONS sip:user@example.com SIP/2.0
To: sip:user@example.com
From: sip:caller@example.net;tag=3ghsd41
Call-ID: zeromf.jfasdlfnm2o2l43r5u0asdfas
CSeq: 39234321 OPTIONS
Via: SIP/2.0/UDP host1.example.com;branch=z9hG4bKkdjuw2349i
Max-Forwards: 0
Content-Length: 0
```

3.3.12 REGISTER with a contact header parameter

This register request contains a contact where the 'unknownparam'

parameter must be interpreted as being a contact-param and not a url-param.

This REGISTER should succeed. The response must not include "unknownparam" as a url-parameter for this binding. Likewise, "unknownparam" must not appear as part of the binding during subsequent fetches.

Behavior is the same, of course, for any known contact-param parameter names.

Message Details : cparam01

```
REGISTER sip:example.com SIP/2.0
Via: SIP/2.0/UDP saturn.example.com:5060;branch=z9hG4bKkdjuw
Max-Forwards: 70
From: sip:watson@example.com;tag=DkfVgjkrtMwaerKKpe
To: sip:watson@example.com
Call-ID: cparam01.70710@saturn.example.com
CSeq: 2 REGISTER
Contact: sip:+19725552222@gw1.example.net;unknownparam
l: 0
```

3.3.13 REGISTER with a url parameter

This register request contains a contact where the URI has an unknown parameter.

The register should succeed and a subsequent retrieval of the registration must include "unknownparam" as a url-parameter.

Behavior is the same, of course, for any known url-parameter names.

Message Details : cparam02

```
REGISTER sip:example.com SIP/2.0
Via: SIP/2.0/UDP saturn.example.com:5060;branch=z9hG4bKkdjuw
Max-Forwards: 70
From: sip:watson@example.com;tag=838293
To: sip:watson@example.com
Call-ID: cparam02.70710@saturn.example.com
CSeq: 3 REGISTER
Contact: <sip:+19725552222@gw1.example.net;unknownparam>
l: 0
```

3.3.14 REGISTER with a url escaped header

This register request contains a contact where the URI has an escaped header.

The register should succeed and a subsequent retrieval of the registration must include the escaped Route header in the contact URI for this binding.

```
Message Details : regescrt

REGISTER sip:example.com SIP/2.0
To: sip:user@example.com
From: sip:user@example.com;tag=8
Max-Forwards: 70
Call-ID: regescrt.k345asrl3fdbv@192.0.2.1
CSeq: 14398234 REGISTER
Via: SIP/2.0/UDP host5.example.com;branch=z9hG4bKkdjuw
M: <sip:user@example.com?Route=%3Csip:sip.example.com%3E>
L:0
```

3.3.15 Unacceptable Accept offering

This request indicates the response must contain a body in an unknown type. In particular, since the Accept header field does not contain application/sdp, the response may not contain an SDP body. The recipient of this request could respond with a 406 Not Acceptable with a Warning/399 indicating that a response cannot be formulated in the formats offered in the Accept header field. It is also appropriate to respond with a 400 Bad Request since all SIP UAs supporting INVITE are required to support application/sdp.

Message Details : sdp01

```
INVITE sip:user@example.com SIP/2.0
To: sip:j_user@example.com
Contact: <sip:caller@host15.example.net>
From: sip:caller@example.net;tag=234
Max-Forwards: 5
Call-ID: sdp01.ndaksdj9342dasdd
Accept: text/nobodyKnowsThis
CSeq: 8 INVITE
Via: SIP/2.0/UDP 192.0.2.15;branch=z9hG4bKkdjuw
Content-Length: 151
Content-Type: application/sdp

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.5
s=-
c=IN IP4 192.0.2.5
t=0 0
m=audio 492170 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.4 Backward compatibility

3.4.1 INVITE with RFC2543 syntax

This is a legal message per RFC 2543 (and several bis versions) which should be accepted by RFC 3261 elements which want to maintain backwards compatibility.

- There is no branch parameter at all on the Via header field value
- There is no From tag
- There is no explicit Content-Length (The body is assumed to be all octets in the datagram after the null-line)
- There is no Max-Forwards header field

Message Details : inv2543

```
INVITE sip:UserB@example.com SIP/2.0
Via: SIP/2.0/UDP iftgw.example.com
From: <sip:+13035551111@ift.client.example.net;user=phone>
Record-Route: <sip:UserB@example.com;maddr=ss1.example.com>
To: sip:+16505552222@ss1.example.net;user=phone
Call-ID: inv2543.1717@ift.client.example.com
CSeq: 56 INVITE
Content-Type: application/sdp

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.5
s=-
c=IN IP4 192.0.2.5
t=0 0
m=audio 492170 RTP/AVP 0
```

4. 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 should be considered a seed to stimulate thinking and planning, not simply a set of tests to be passed.

5. IANA Considerations

This document has no actions for IANA.

6. Acknowledgments

The authors wish to thank the following individuals for their participation in the review of earlier versions of this document: Aseem Agarwal, Rafi Assadi, Gonzalo Camarillo, Ben Campbell, Cullen Jennings, Vijay Gurbani, Sunitha Kumar, Rohan Mahy, Jon Peterson, Marc Petit-Huguenin, Vidhi Rastogi, Adam Roach, Bodgey Yin Shaohua and Tom Taylor.

Thanks to Neil Deason for contributing several messages and Kundan Singh for performing parser validation of messages in earlier

versions.

The following individuals provided significant comments during the early phases of the development of this document: Jean-Francois Mule, Hemant Agrawal, Henry Sinnreich, David Devanatham, Joe Pizzimenti, Matt Cannon, John Hearty, the whole MCI WorldCom IPOP Design team, Scott Orton, Greg Osterhout, Pat Sollee, Doug Weisenberg, Danny Mistry, Steve McKinnon, and Denise Ingram, Denise Caballero, Tom Redman, Ilya Slain, Pat Sollee, John Truetken, and others from MCI WorldCom, 3Com, Cisco, Lucent and Nortel.

7 Informative References

[RFC2396] Berners-Lee, T., Fielding, R. and L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, August 1998.

[RFC2822] Resnick, P., "Internet Message Format", RFC 2822, April 2001.

[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.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.

Authors' Addresses

Robert J. Sparks (editor)
Estacado Systems

EMail: RjS@estacado.net

Alan Hawrylyshen
Jasomi Networks
2033 Gateway Place
Suite 500
San Jose, CA 95110

EMail: alan@jasomi.com

Alan Johnston
MCI
100 South 4th Street
St. Louis, MO 63102

EMail: alan.johnston@mci.com

Jonathan Rosenberg
Cisco Systems
600 Lanidex Plaza
Parsippany, NJ 07052

Phone: +1 973 952 5000
EMail: jdrosen@cisco.com
URI: http://www.jdrosen.net

Henning Schulzrinne
Columbia University
Department of Computer Science
450 Computer Science Building
New York, NY 10027
US

Phone: +1 212 939 7042
EMail: hgs@cs.columbia.edu
URI: http://www.cs.columbia.edu

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 3.

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);
```

Figure 58

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 --
H4sIAAAAAAACA+xdSXMbSXaW1O5lOOY4JvrmU4nRGHWLLDCXWkGBTbZETbO1
tEKk1BHjcaiTqCyigEIVWFUgRDks2zpMeC4O3+3DhMMRvo5vdkyE/RfUP2IO
PvnoqzOrsNSGhQsWtSq7IRCorCUT7/vyvZcvXx4Rg/htWisbJLg2owJYUSQp
fFcVOfHOi4zRNci+kjGUZJnVgzKQwTUBXJtD6fgB8QThmtfw28Rr+qPqTTre
a8vg/R0p3z453P/28YHgW+1Kx6feDn1JWm2bll3vWDjYf7KJymB15blFKv1P
m8/uPRHqrh9I5X7dmtuqyEABW0cecWr16iu9/kvp6EHTaHQkjLC0uvKIvBTv
u16XeIZfEVR2yfseO0lY+4ra9oawa7NLOQb11oQ7/ElI+bh8xI7sxO6wvRWQ
46qEsYTw6sqhy07+jgS+62wIh3W3RXx2btiMoNwNv080ZXt15S6xbXH/XkU4
6su8b5iAPZ6DHMMHBOh6B+E6rjtOFwC9ZjQdyu60W6vRdlARSLttWzUSWK6z
6Rttdr0DelIRsI5YE3Wh15GrK3ZFYM1bXXk3fn/WF9GPNkMCmIR/CShp/COI
CvwvFP8McUP8c7Tl1ejDmB+rMXgl+SMELMZIQmkCwDmUAnX2VxmVYYpFxiEw
huieFPvEcB3TRdiSEGicAuIDRismbmCnj1jtXcXqjPDPgE9nOfxPwj8ESE7h
X1LUYvyfS9l//Hz/cO/q4e/QIIQ/YuOrzgfsrAYQR28og2wYZWN7y2EYZoBF
2G5ISBR9aphOK9QiIvxCNuTqWBKiRx+hnMR1kxy1pLu6co/dsiLc96wNAUDh
G+IIiImiAJUKAOx/Ye/gkN3RdQJSY9xzJ9ZGfn25HGvpdlSROoF4eNameUzV
O/yQOsdBnbVBhpx3TqusH9xqq86UH5ueCUhXmSqhIpU1UGcNFPafSANilFdX
/Kq4ulKrZg8EVcCZrFUlHcNyBdblUAXC08Mnm7vPnwisVYgfPLUM6goYIXVw
CLPnIFUvaLdIu4Kh8PDJ3feZDt9L/ndmyv5T2H8Qp/kfy3LB//PW/3INp6QN
yEucamvT2IDdNPuHZWgEspIyAydbgdGgFJakFRidmteUlAnIxJ7JG5R3Em2Y
aO+xMsLke1fxbzmnAC7S/6NiGMO/zPGvQqnA/9Lpf43yBTRAiCWoMI0eIknJ
qIFJSEaSCOoEWL5jkKZvNIjPdEHuoyH+0H4bqfkNbEh5a2tja2MjpsCtfePS
nnep4dIEL2yxkqei4Ylq3Tk1ODhShYOL0uFYr59Sz1+k/QckCWfwj5UC/0sz
/quZ8V/NG/+nHfoTI/9uORz8J434J13qWidtGhv3D8u9oX/qIT+UdD7oq+lB
fzC2w/fMOXRUI8xYXqj+DwHMjP9YLcb/uZTe6CkgAIRvH4wbUnUthW+IUeTa
```

yZyDZLkce6XP88kR9fiJQ2yGUgiwJmkSQACyAZ8N/iZp4gaiXpcapr97YJiD
ORd5oAIM1Y+0ahIpH5ChHffNhUw1anR6Xiqk52kA+lSOnaGDKLw4dw8hLc4v
2+fXFHRtpKqga5sQqYXPpyhXwf/0ZTBb628K/48qp/lfQrDg/+Wb/2u8uMgU
gIRo4KUVQSVO/5EQJgy/1ZWn9KRjeYx0HTeoW87xQafddr3AP6xb/kbel3tW
UKfsRk889+WZGDudf2FRv1e3f753z911zrLHcuYJEyPc4cMDwXRtQ2RELjLp
CQeSSVpwemxZFs3yyDquucZC5/+ADKWM/Qdwgf956n8S0iVdUTGADI1BQD0O
MOHIo6QpMFQJHq1R65Sja7SGCNKaHsK6xuDs0jjWewJHfMM1caeBAGZ/OiZm
T9Bhf9VN4mNaDx0/xNetoc6HJV1K6X138mgrMhixThzftHqa35081S+qqAPU
cAynSXEuSkcoeKyxGQ3vXRz/a4yxvQXP/6gKzo7/hf23PP7f5MitXdYjrPN5
fvaPi0ZNsHecY+8sPcPeZ5BIZBPKQpdahLGF0cRTOImT0/fZEVvk8zuSiLCM
8Lkn9nVWLuAXHuMYnq1nuMaEmrQWO/8jg+z8D1AL/M+jPN375f7B4d7TEK+5
4M/gxydBx3MuNPE79Pzy26X8tX2fzb2m+fy40fSCR11CvQcP+k7f/FPixNCX
ZSY9EOxkn7NPDkjoNztGQPzy61BXERNBxMrOcRfGKWir4zQdt+uEN/nReId7
fYYWiX9JUTP4R4X+/77iX8MaGyTPh3k0EfM4D/N3zgX67R/hnJBxZHv0ZKH+
P4jlDP5lpZj/XRL8T6fqZ4/2ArWQDBu4gVFmEZC2urJfEXryl9DmdZ3b/qZj
Y7uJMJYgSg3TvZul43yHuv8Q6qO9FUjOY6wwXHmEu4498NBaigeRJPqr/4zJ
49MEykgQ47xuGpBdurN0VWW40kM07cRDSc4TKJPTCVgDUbz17IOaryAkRpja
8ilmuTKF+rVZT/9Mnv+XQJr/ESzif+ft/2Evv4R3ObWWJBDjrjhTZceFkir7
tKSiPLYL9av9EmKMdUrZG+vCGvXj1+v5YLGW4T11dcWqCJF88pUhQA9dws1G
00HYddgHIOLYshDEF4ZK07p9OOnmT9SMmeRfXREGfqqScjdvKcgW+551xpZT
UmCLVk/Zm91hTZdLWCphuH1FrDmSNIuwgKKcl//Rgv3/Stb+Z0gu+H8++n9J
UpmuXJLkyAbw6LHlBx7xyiOtgbXSr1j1XjC1R32349USyu72YHn/xJoh/ZsI
owYxzEYiKCwSTU76zknXxZJ3grCFpYbH5wZrjo2w49lNx+dxYckJ+rsT1wAi
oJfMV75D7fgIomNJEuI9krtsMeE8ILZFfBhNCaYmBO+WlPuZqigKD0tXzV4T
hxVxquKV+x9YJzsd217s/L+U9f8pxfr/JbP/uZiIJQBE/sco03ZkpZ5nT9dR
lL9jzHLgvkhiHQHsGOZps2EQ32ye4K7kBt4JIIZvMlIgp8ZgSbCEdY1rf2MM
/0nTfrFAnSEUWVN2Midu59QaVfFhZbn9hayzvY5nLXb9fwL/SoT/wv83b/sv
7cD78qnbCWi1hO+muKGE964kNwCWmO2j5a0JG7XqPuSPUXEBfVEOjUWR6w31
hkgaFB95IuOGE6BrXdEkXerV0YlYlxx5SB+Matjgj8Zbj6JrikGdinWmxkwf
8TdL51lhBhblcsVy/I65YPtPUmCa/yVc+P8Wzf9Dju+xZBi3NXlOQc+ZUyiH
cjaRFO3esttz8qA+chJBL3KjTMJ/0KJBfaH6HwIZ/KtyYf/NpdxkAkA96geW
cyw+YpLgGuHyO1lRNf3F7fXvyyXLMSg1br0OSQK+6Dgdv0Ps8rOn+68/D1zx
iN70Ox794heW79wK1q3gs82aR16dfbmxubV1u/ILK1ivE78KN7rU8oybt9vE
/6zrvjZeBK5b/txwKT9NtIIvdiaHn/Q9O3Bc+Gb5ZklRbr9Y//7W656/6qu9
h5Vffyw8fsbergn3+Ke/6TmlztueXDdX4DapA19/H/2Bbq2/iP7Ct0s3y2J0
o1bHMYiTcn6Z7PQnPLTk1q3X67dfsMql6toPf/v292///e1//fDm7e9/+Icf
fvv2P97+51qoML8Ir/oa3lr/nl1ZLMeU3z6Ou65nlH71QLxZZj1wa4d//P71
F59v36n8enPtL//iy7/+q1gaLYgQ1mThHDKQVtYRD0+lLxmn+4zEv6bEoB67
d+n2+gvW/ZX//cN///Ff/+2Pf/fP//NPf1jeNTDvt/53imQJL9T/p6hKVv8r
4v/mrf89Y/rfVzvTxQBaZnDcLY+a8V2HGGBZliErO6xquWZbDPjJwDp2t2q7
7jp0my/1qzGiEkOHQ+8amafZahHD8Kq+n/az9z0Q61CRQT+OL14tebsEaUbC
D1Wo5j1nLKZIVgaK79XmhLlEVr+rIE/WBZ1g0fN/IOP/k5Qi/+cS2X/TZsG8
1LIgLXT4S2PnBiJxTWaIMnzTavAs3tgYxoLIkjQm/M13W3S6LKGjwd6LDDZd
r0WCrHIjhdrNnRC5rGeEO+1aqyPwNcbVNTa8rW2yL+9s9g8v7Pe3Xed41gHA
E/CvKKqUsf+K/C9LhP/QiNoXeBCXQAReT3BIiwquKTDd36Mtesk3oe25HBoM
V2u5a3orCuvTRDA+rJ5S74yL71W+TondoVv8r/AmvJWzflX9Omt5dONZNCnv
FdqclBl9j9nbd1ZQf+w+5w/ALdqIo0mLvGLmoB0+TkTY/GFn+XU2IBrp2jL+
LyGpL4r10OKdiaxUo1vE7hDKyNX803/+8Fe3qe8PmnDlf8RG7/5gw7Rwj7Iv

z+b/b1/krIGugDUeOQAGysKzqGfEh6zmVb5EjrWK0Ot48aohLUa/6+yu3+5z
hhhyf3UedwrblOMDZAyFcdJMO83WQMkaz7M1YKpG3n1Aus5upg7SU3X2s9dJ
ZgTjdXKuo6balPM8SEnXybmOnKqT9zxSuk7OdVJ9/FyoCJk6qV4+zauT7mdB
yN4LpK+TrQP1zO+VU0nLVsqppU7+6aEyRR15ijrSFHXwFHXQFHWmEGkIJtfR
J1fRJleZopOn6OMpuniKHp6ig6fp3wlVMlZlL3GOUR34dfqG5ryUvXm8oib1
6HpizOwCdMsL7Nlhvw9BOXZwHMw6AHBi/h8gp+x/GRbxf3O1//MTaV1NlJ+O
dJiZN4wH+g9kEO7EnN+9WcrJ8SbyDLfpUcAVO/SZiOsYaSpYEiKwu75h+e3F
5n/Pif8o9v+aT7ma/f/WIsit3cnhgWFGvglx/31R5KEH5Khm7JjMwjzLm4xT
wIjUnJwWMmflO/aLrf/6+F98/D9Oz//LABb5vxbt/98SbG8yBUTQfmlSUWeD
mKZh8ZVuTpMVGEeZ2TPzfXKSEkLpjJKBGi51Lb71Uhjc3/Wc6C8NS14zXFTU
m/yDmKmUioSniFWFI1MXIayPUyzg/BWL8YbFsikWRXlX+J8TYLBQ+0/Faf5H
oIj/nSf/5+t/IzPAXsog1DQoJTm+J4Fm0zlhuh/jeAkDXScGX/pNuhTv5OiA
UNOxpk3I9wEn6YFY7+d4nQnLL7/7qFWzF53/B8jZ+M8i/8NS2X/nTqbDUykg
PB1NuHzXhjy3EV9uFCOKSFTNusNVQNez6qRrGi7BruTJyMWW5w93CIIyU4aG
RmIGmBlLVAHZWnIa8AF9GWy2bWI5HNaH7LHpLV9wXKFLzoTAFfh8KOuartAi
zpng1gIa+ALxqODz3SV8avBKR1TgJ5aXw/hsWX6LBLX6LElgEv6xLGXG/yL+
cxn9PxeK68SSjORx27/EZbDBMw/6Lw3TnjaVe+HquRL8o8XhX2XK/hD/aoh/
ufD/zKU83vvu0d7h19/eWw4GQBMYYKSmnlLUxzPF6OR/k2eEsX4RlT5Pm18K
H02rYwfWjC2Ayet/pRT+JQUU+78txP/LAN1myuuU6z+QPAl3fXnPz7acOx8U
rbcaQn5AEj1R1TW+TdRODEm9U/Spz0FjSGvg0+buaZ6EeLQN0z+SsWCm8X+H
mbD4OtvpAlHGOaOxkvoZMh71czIWGjmJja48oUKhly2yODV7xt6fKfS/RP6v
kP/Zh4L/Fzz/N8r7i2Tpsvpg6NrBcS8wl8PEsj4EQxeSJ0uoiwfhQGCKcCA8
TWK/5JPJ+HKOXvEiu37xRDcjIoXw3EIFHdejxHedhe7/gzL2HyzW/8+n9Pf/
ZP0vXHpvz4EwEYZivrOnxbPz6jpCPI/nyN3bi108Fzj+u6fU9muLjf9Dcnr8
51uCFvifp//XJe3yEaXtyubmEPS4ggFAVxMGrGUifXCCOiI5jCJ92OXBCWMY
pxsF/jTtYZZuHKUQHrU5d28Nhj7NzK9U5CQ66bjBEVls/m9VUrL4L/y/y6P/
h+v/H3ll4ZuywPPyRKNoVvXfnmrz35xNrmA8DrgvkqEZMC62X08r7dPueiXr
FRnIYApamDrQAy9i56sr8BF79Jh0gsXu/8uT/aTxLxX+37mUGe//p6kYwXoD
YYi0cTvbDcQw4QPoQ5pxxsg9/fiAP3ROjrD9dztB3fWsVyGEK8Jj91uHPnDc
rn9Yt/yDWp2nIHHb5KRDRYYDUuUBGhmU/yhVA9bxR8SoBYvFP07b/6AY/5cN
/+PQn4t9XdcwRmOX/Aykr4klmfiejU3j6HQHgjL/Dw4XAY7b0BPLZaix2pi/
8Ji9iBN7iI7f7YC9yskdD36kYSTsF+D7JiwW/7KS9f8V8Z/vOv61SdCPBC8F
/fSs7hVu7fNohKtxEvTfgW18Llx8ZlARb7ZbAE60/3Em/kNFBf6XBP/5CdjR
OAcblpBvmO5ot33cVzDG+4+4NY4a9UEgGFY0HUt8AQkzEHUIMDscI4YBuQyk
GmEXtA39lDj2Ce46nu3Qrms2iK93rKxuggFr797LtuXR0CkxhzJ6aiLZwzyF
qLBFo2erIg0gXZV1RcFIg/Il3JhRT9nHi8S/LEuZ8R8U8/9znf+TARYOqHdq
1ajwzCGnxLLJkU3PCX8Qwv/Vq5eGxXfs9LtTE0AmhKrHAJT4RsOkje7AF4DC
hSEallQVYSABhDGML/VIcgCTbMclPgWuyeNK6w46cU2X8A+I0K7Jr+XxBOSB
dybumgH12C0kHQOkYY11Cn/nH/l9dI2pNN8Rz7GcY74Ajd3WPaVe0PGosLbv
CE8895ih0197h6YVfKO98PVfqpyJ/0Fy4f9bLv9/6OR7kTUCRq+dTMeGTpES
YGwygEhUe65BHhxkMG4wVld2azXaDnqrsxz3yDXOBr69c0wKQHl8THhiUefS
7AFw2YkAn7asBef/AHI8/7cCiv0/51jS67+2WCOrnZIEYuA8PymMhLrrHUfR
ABhJaFw8QF8uc+cDtOFw3wd/CoIbiS/azZqvii2rRTe4Gh2VMDSctTbYbFkv
qbER+8K3jh32TawuG9XJMWUH2JVjH0yPHI+jlOVfj+bzXlns+g8VxPd/jPAv
F/t/z33872UqvdgScA7spLAzNd1zotidmBWQd5OkJeBTx0gyxsAZEK4Mo1Yn
QRQ9Ae64J+wIOXWaNjFps3GCJasj4Q7p0kFiII3r8GwEHSgDkzKV9p4l09Tt
rFqAYTZuIM0oW0NGiUrbcwO35trVtTRdiarIzyHcuFjLnNdiVe3jql8nMHPs
yOU7zXln1TWRlb3dr+5jbRfs7u5pKpDuMnzdh0CC2le7GlDv3l/jRCSK01TN

tG7Ig/wihSRdUJKWYQg7b7yLFA93j9q2LGlNLirO4xC4Fe53Ul0LRaTcVuMm
/qHHQ5SpJ+45NdcIXQNHlsMAOKxyz/Lbrm9FwQckYGJRb7HvtwTTsmn2yqsr
4M0nf//RT27/5uvf/N/q9Y9v/O7NJ2fsq86N69fhT8FPPvpw/Wcf3PjzD6+B
nw4rXf/dm49egTcf6ODNjX/53Qc3rt+48cm1f7z22xufXbsOVgf1rn/ITmuH
V/ng2Z989OmNZwfwU/Bz/uGTTzkWLNP1HIvAn4M/419+/OknB8T5xvUp/Bn4
U/7Nyqcf/n97VwMcVXWFs0kkGsCoZIhtknpNcAJpNtz33m72vQ3pxEIdpJgA
SYCxjPXt7tvsy27exveTJQyM2aSVij+MmlrFH8iIAjo4FopFZiw/7QAtdbqL
Nv0xSbFMlVTRAceKaZXe+95usrvZTYKtG8X75ffd97P3nXvOveeee865iG1E
lZoHy/Dx9FkN+LBRUNSFgqyKbkxEIeJxoXbAbxXMhAwFaQrv6kdbby+YSaFD
io0cQsf/pTKF8Bv4OCv/urFSD4NbYylguiIjK/hABgzemxkMZpzozpyWd93x
YXPBPQ9X9Cr973YeLCzc9PESaoaoPbtyhzPv5VOL/jLU0//Glosd7b9x/Gnn
O3UnPtlY+uLBskfPDjicq9b7vh3oXw+nN9Sc3ffvB/d23d33nydsr2wb7Npa
eE2wj965e+tLufdcNffM5o2nXuk4IMxbcqzvZHjT2uPe+h3MxrrSzCxThumZ
4BAMvg2L0RsUX5s9G+YHk70FJlLxrOxMpGPoV16dPTs7v/Zi0YzC/Z6luwq6
n2ob2nBh4Ad3hmFwCz5fmh18GAY3d+Z7bynI/2dP+02djdKsxr/2nm7M7325
V90up5UPukwZCWyYhajv/+yb1WvO5Px6oOLTfw2tvfvWoh+9+XF/uHhP386f
bJh/+nzlk56ex7/b9Y/cbX/7rOjVuiNHMgu6bziyuyrv3KHB4rfOtF1TMfz4
+8uWbjj0Z63ciUi7eGBweLDpaGj5udca//DO4Yayex3PLviA368d/nSBOiTP
qThgovbuP5pz8Kd7195YsWvHHURgiMAQgbkEgaG6TH2wy3QSDUVwfVrpESOd
oyPgtuDP4fUjRLoqi4odEWHR6JkrqDwkVhZooyw0ZKqg7XZYOno2m0KcMRe+
efbY6gO7NBPLLqLOnAv+rOA4XDF6UR71PbgQ5k670ijIyoFXj/yfmZmZ0Yma
a/TYVAtzjGrmxJfPTRCt7GDnnRczZhcNH3775tBvHxl+reepI9wnAx+VXbA8
994F276Tx3658sPcsqaPPljXzSyXb4DbuXk3ntph7Xl++vk67+uPzXIMbj8/
d9GJZXe8fvTQ73f/cc404ZEXztaWzoFNv5tx6sXuc6zn708/dPr51ff1X7un
dzOXcf2P31rbscxaKm7Z+FhF6+I3Hto0OYXJbCbB2pcdVCRtX3QA+IT+X7H5
/wz7D20j6z/ptv/wPl6qbeEVf6sYSW9RD9B0JbHUmGDLvK/NE2/0vY1fbXbX
y6v4Fa4GO0BNHuvgrfNZnDE31tVrdce6dR1mGl3mERWAvlWPACSt1YGmlWgG
BYwIEeMqxj4y6dYL4hw1UC0E3lifdfv9wIGaFjUbrrXfjw9k/U+SCS072TUd
/CLo+TV64o7LQP7x7BVvvalMYf6vKluC/FutxP9zatZ/Pt/+D//Tvg8ghge9
YovHwntdEn9XizeAVHqFV9x44SeFkWxSO0JMnAY44aaGhY3oLnq8u3y80uLy
uBP9Y5Y2AJUZ5z6aY6EmufAbxVez7vt19avqgGoZ52ZecUE3I0hJnHLGdX+F
PCe63JIgJFl6UuWAMtXjvxUmyr+FgUT+0y7/stDqVwUzokdz8kVfAFL5g+nc
l4LjsasG5OJN9yk+KN5+7/OjLsUsC4pfk51Cso6Fs9AcZLlYPQPzs190aoyF
s2IrvSS4cQS5R6YZi1WesCOhGZaLDS5Pbte/fCLGNcmrOFu/2B5gQv3fwiTK
fxWR//TKf4LflBETaVf9Kt402QiNdBqMn45sEBGulEg2iHTIfys9xfLP0GP8
P2gy/08LJhf/JSoOyY50V8bG0ayNGRmlPaqKE8YEAoHKZGo/EsxEqdZ5TZ/+
ezraPILEq4LLnJjGe3QwtnAwNsBj7GicOuEoNZlI0AV49dW+xO+RftjQKqqe
73zdsoVr0pTnf6MgnTj+WymGyH9aENXsaQhBDaDLyxlQDqzl5TQIvRr6FQgH
w13oT+gXoX2hPeHNuCyhKNwVvh+YQehguDN6LrRvPJ9Ijk2QSYqhLRwd31FE
mBJvBenFg76XaaEFOSC43MrNDeOkkksZjEpRxnZzKbQWwaUZrkE0zSVz+eYm
5SKTLIyLY8dkmLu0NDEcm9J1hmPnU7Tty7itNMFXBAFFlNqnNv6HslnH6H84
JQDp/9O7/tMutgve5lqnhxdls0uSnXGTJqS8ocmhhGjAxyqH9cA+M3fi20E1
AAB1sNgTsQZ1YBSLvxhGmpnrlvFOc8AOSpaAFX5FkByC3AzWrFlTUhJxXTSy
jblkfLJ2TEws/oWfXANwkvcWFzvBSpTB8UkWoeJ7fb1KeqyPU9EzD0PI4Y+K
iSPCVyBKADAfXayTA+ChzvC5HNFDo6Mdw0Elon4qmGh1QuAWXmr2Ca7FAu/C
cYcASELAbZSBdt6nCehReNYtShoqGXuyyWgT4/5VSH1tkjRF430r8Wk7qK6u
qEA/1ROOXXr+A9yOEVdVPQxUiWt+nxzlgAD6nBq9AtESs+Q36wWoQdrtIEoT
3WA4H+A5OIbSJuLhLo4rIs1nUAhzBgBRn1/R28ICgB1YR0hsPBH9i/UJncBU

```
FVtJW62VaHTHz4o8qAbdFI1Ex2GbaPCzlyzX/GiqARRVFiXEXiWYvVIyFnoU
orbB6zVRH1pUor+lUWcF20NcxiXV4C5Ud1jJXHoWuJQ5oC3pGdbXCUj83FM8
/6fhWPsf0f+/Qut/41n6mj2Ky0LFdMARlmvBpjyfW2ql/bTPwshWDRrmvjjT
HkNT46zsTWqPT2xFEBNtB5CkfiUgICAgICAgICAgICAgICAgICAgICAgICAg
ICAgICAgILgM8F+lVpDAABgBAA==
-- END MESSAGE ARCHIVE --
```

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.