

Network Working Group
Internet-Draft
Expires: October 13, 2005

R. Sparks, Ed.
Estacado Systems
A. Hawrylyshen
Jasomi Networks
A. Johnston
MCI
J. Rosenberg
Cisco Systems
H. Schulzrinne
Columbia University
April 14, 2005

Session Initiation Protocol Torture Test Messages
draft-ietf-sipping-torture-tests-06

Status of this Memo

This document is an Internet-Draft and is subject to all provisions of [section 3 of RFC 3667](#). By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she become aware will be disclosed, in accordance with [RFC 3668](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on October 13, 2005.

Copyright Notice

Copyright (C) The Internet Society (2005).

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	4
2.	Document Conventions	4
2.1	Representing Long Lines	4
2.2	Representing Non-printable Characters	5
2.3	Representing Long Repeating Strings	5
3.	SIP Test Messages	6
3.1	Parser tests (syntax)	6
3.1.1	Valid messages	6
3.1.1.1	A short tortuous INVITE	6
3.1.1.2	Wide range of valid characters	8
3.1.1.3	Valid use of the % escaping mechanism	9
3.1.1.4	Escaped nulls in URIs	10
3.1.1.5	Use of % when it is not an escape	11
3.1.1.6	Message with no LWS between display name and <	11
3.1.1.7	Long values in header fields	12
3.1.1.8	Extra trailing octets in a UDP datagram	14
3.1.1.9	Semicolon separated parameters in URI user part	15
3.1.1.10	Varied and unknown transport types	16
3.1.1.11	S/MIME signed message	16
3.1.1.12	Unusual reason phrase	18
3.1.1.13	Empty reason phrase	19
3.1.2	Invalid messages	20
3.1.2.1	Extraneous header field separators	20
3.1.2.2	Content length larger than message	20
3.1.2.3	Negative Content-Length	21
3.1.2.4	Request scalar fields with overlarge values	22
3.1.2.5	Response scalar fields with overlarge values	23
3.1.2.6	Unterminated quoted string in display-name	23
3.1.2.7	<> enclosing Request-URI	24
3.1.2.8	Malformed SIP Request-URI (embedded LWS)	25
3.1.2.9	Multiple SP separating Request-Line elements	26
3.1.2.10	SP characters at end of Request-Line	27
3.1.2.11	Escaped headers in SIP Request-URI	28
3.1.2.12	Invalid timezone in Date header field	28
3.1.2.13	Failure to enclose name-addr URI in <>	29
3.1.2.14	Spaces within addr-spec	30
3.1.2.15	Non-token characters in display-name	30
3.1.2.16	Unknown protocol version	31
3.1.2.17	Start line and CSeq method mismatch	31
3.1.2.18	Unknown Method with CSeq method mismatch	31
3.1.2.19	Overlarge response code	32
3.2	Transaction layer semantics	32

3.2.1	Missing transaction identifier	33
3.3	Application layer semantics	33
3.3.1	Missing Required Header Fields	33
3.3.2	Request-URI with unknown scheme	34
3.3.3	Request-URI with known but atypical scheme	34
3.3.4	Unknown URI schemes in header fields	35
3.3.5	Proxy-Require and Require	36
3.3.6	Unknown Content-Type	36
3.3.7	Unknown authorization scheme	37
3.3.8	Multiple values in single value required fields	37
3.3.9	Multiple Content-Length values	38
3.3.10	200 OK Response with broadcast Via header field value	39
3.3.11	Max-Forwards of zero	40
3.3.12	REGISTER with a contact header parameter	40
3.3.13	REGISTER with a url parameter	41
3.3.14	REGISTER with a url escaped header	42
3.3.15	Unacceptable Accept offering	42
3.4	Backward compatibility	43
3.4.1	INVITE with RFC2543 syntax	43
4.	Security Considerations	44
5.	IANA Considerations	44
6.	Acknowledgments	44
7.	Informative References	45
	Authors' Addresses	45
A.	Bit-exact archive of each test message	46
A.1	Encoded Reference Messages	47
	Intellectual Property and Copyright Statements	52

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 "abcbcabcb" 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:

- o line folding all over
 - o escaped characters within quotes
 - o an empty subject
 - o LWS between colons, semicolons, header field values, and other fields
 - o both comma separated and separate listing of header field values
 - o mix of short and long form for the same header field name
 - o unknown Request-URI parameter
 - o unknown header fields
 - o unknown header field with a value that would be syntactically invalid if it were defined in terms of generic-param
 - o unusual header field ordering
 - o unusual header field name character case
 - o unknown parameters of a known header field
 - o uri parameter with no value
 - o header parameter with no value
 - o 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-f0rWaRdS: 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:

- o 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
- o The Request-URI contain unusual, but legal, characters
- o A branch parameter contains all non-alphanum characters from token
- o The To header field value's quoted-string contains quoted-pair expansions, including a quoted NULL character
- o 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
- o 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
- o 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>EFBBBF5A4A7E5819CE99BBB</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:

- o 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.
- o The From and To URIs have escaped characters in their userparts.
- o 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.39203ndfvmkjdasfkq3w4otr0adsfdfnvd
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:

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

- o The To header field has a long display name, and long uri parameter names and values
- o The From header field has long header parameter names and values, in particular a very long tag
- o 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 contains extra octets after the body (which in this case has zero length). The extra 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=z9hG4bKklasjdjh
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 MESSAGE 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

MESSAGE sip:kumiko@example.com SIP/2.0
To: <sip:kumiko@example.com>
From: <sip:fluffy@example.com>;tag=2929017b
<allOneLine>
Via: SIP/2.0/UDP 127.0.0.1:5060
;branch=z9hG4bK-d87543-5032442a6f48352f-1--d87543-;rport
</allOneLine>
Call-ID: 74dd6bf53ebdf741@Y2ouY2lzY28uc2lwaXQubmV0
CSeq: 1 MESSAGE
Route: <sip:127.0.0.1>
Contact: <sip:fluffy@127.0.0.1:5060>
Max-Forwards: 70
Content-Transfer-Encoding: binary
<allOneLine>
Content-Type: multipart/signed
;boundary=4d7f63e86a96c361;micalg=sha1
;protocol=application/pkcs7-signature
</allOneLine>
Date: Tue, 12 Apr 2005 17:30:54 GMT
User-Agent: SIPimp.org/0.2.5 (curses)
Content-Length: 1567

--4d7f63e86a96c361
Content-Type: text/plain
Content-Transfer-Encoding: binary

This has a null in the body.

--4d7f63e86a96c361
Content-Type: application/pkcs7-mime;name=smime.p7s
<allOneLine>
Content-Disposition: attachment;handling=required
;filename=smime.p7s
</allOneLine>
Content-Transfer-Encoding: binary

<hex>
308204E006092A864886F70D010702A08204D1308204CD020101310B3009
06052B0E03021A0500300B06092A864886F70D010701A08202D2308202CE
30820237A00302010202080195007102330113300D06092A864886F70D01
010505003070310B3009060355040613025553311330110603550408130A
43616C69666F726E69613111300F0603550407130853616E204A6F736531
0E300C060355040A1305736970697431293027060355040B132053697069
74205465737420436572746966696361746520417574686F72697479301E


```

170D3035303431323137313933385A170D3038303431313137313933385A
3062310B3009060355040613025553311330110603550408130A43616C69
666F726E69613111300F0603550407130853616E204A6F7365310E300C06
0355040A13057369706974311B301906035504031412666C756666794065
78616D706C652E636F6D30819F300D06092A864886F70D01010105000381
8D0030818902818100978BB1A8AE41595A61730F9DDB218A7EC4F30F33C8
584E5B7B063FCFABBB8F95BF1C3F763D85A70F1CE95374CE801CA74D333F
EA7913548BD8CA4E683C0495F6127B365B067C8A04EEF31D1E4DD7A99492
2DF15F468622B49B3DD3AACEF5CE36DFAF51FDBE68AAFBDFFB50274365450
DF147F544BB5F2BE3BCEFB39291B764EB0203010001A37F307D30510603
551D11044A304886167369703A666C75666679406578616D706C652E636F
6D8615696D3A666C75666679406578616D706C652E636F6D861770726573
3A666C75666679406578616D706C652E636F6D30090603551D1304023000
301D0603551D0E041604146EA02972AF7BF578F284A873F7979DB3AD59EA
90300D06092A864886F70D01010505000381810010EB632291EB8560CCCB
C8FF9EB8481B8083A7AC2520AF79A4852CE997A0BF81272B8DBD5EB2CB72
97982BF66F20CB5758D85C9C90426571269D4523FA67C86BD72BF94A1C34
2054529F92B370D1C55A1A75F4BC17423B2BBB53CE289AEAD505925F3786
74A5BDAC658CFD48D8C600F21B9E7DDE5A1135E1FE273BA75392DFAE3182
01D6308201D2020101307C3070310B300906035504061302555331133011
0603550408130A43616C69666F726E69613111300F060355040713085361
6E204A6F7365310E300C060355040A130573697069743129302706035504
0B1320536970697420546573742043657274696669636174652041757468
6F7269747902080195007102330113300906052B0E03021A0500A081B130
1806092A864886F70D010903310B06092A864886F70D010701301C06092A
864886F70D010905310F170D3035303431323137333035335A302306092A
864886F70D010904311604140E5123C664D47E6B9BBE2630DD21D2496580
11B0305206092A864886F70D01090F31453043300A06082A864886F70D03
07300E06082A864886F70D030202020080300D06082A864886F70D030202
0140300706052B0E030207300D06082A864886F70D0302020128300D0609
2A864886F70D010101050004818018EEA35F807FF7C7F96B747722367C9
5F72FCD74140721B36BF638375D5DBF13EF4F00A71DC7E3E866E1D51C14B
C91AC2FF8369FF56F2F16603701570A9273355A8D5875718204FABAF5DB7
480A718560B89F1E28F9186431BB2B2452D45DFC68C0EE42DD808C70355B
4BC18F7954DED86ED20AFD57275A6A1DFAAEE607C482</hex>
--4d7f63e86a96c361--

```

[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: 155
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
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 header field of this request contains 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 actual length of the body.

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

arrived over a stream-based transport such as TCP, there's not much the receiving could 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=93942939o2
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.

- o the CSeq sequence number is $>2^{32}-1$.
- o the Max-Forwards value is >255 .
- o the Expires value is $>2^{32}-1$.
- o 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.

- o the CSeq sequence number is $>2^{32}-1$.
- o The Retry-After field is unreasonably large (note that [RFC 3261](#) does not define a legal range for this field).
- o 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
<allOneLine>
Via: SIP/2.0/TCP host129.example.com
    ;branch=z9hG4bKzzxdiwo34sw
    ;received=192.0.2.129
</allOneLine>
To: <sip:user@example.com>
From: <sip:other@example.net>;tag=2easdjfejw
CSeq: 9292394834772304023312 OPTIONS
Call-ID: scalarlg.noase0of0234hn2qofoaf0232aewf2394r
Retry-After: 949302838503028349304023988
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.dj0234sxdf13
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 in the start line, and a CSeq method tag which does not match.

Any element receiving this response 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.dj0234sxdf13
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: 155
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
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 a url-parameter in any 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.

- o There is no branch parameter at all on the Via header field value
- o There is no From tag
- o There is no explicit Content-Length (The body is assumed to be all octets in the datagram after the null-line)
- o 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 final detailed review of this document was performed by: Diego Besprosvan, Vijay Gurbani, Shashi Kumar, Derek MacDonald, Gautham Narasimhan, Nils Ohlmeier, Bob Penfield, Reinaldo Penno, Marc Petit-Huguenin, Richard Sugarman, and Venkatesh Venkataramanan.

Earlier versions of this document were reviewed by: 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, Bodghey Yin Shaohua and Tom

Taylor.

Thanks to Cullen Jennings for contributing the S/MIME message.
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 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, 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 -xvzf -").


```
#!/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+xd23LbSHqei1RSUZxeAVYt1z02QDW6Gyfk9Ehja2a0Poxj
yZ7M7qY8TaJBggQBCg2K1lNxr1IVW7yAs1lqnK7uUtqq5JX8LxBbvYiV3mF
dAM8gABIQrZ48Bg9Q5MiGodu9vf1f+q/G8QirE+bVYuEn62oAF40jKN3XVnN
3kVREfxM4V+pSMGqyusPKlDBZxL4bA1lwEISSNjNqYf1SdB18+ot0z5qy+T9
IynfPTs//e7pmcScfm3AaHBEX5Ne36VVP2hJZ6fPDmAV7068dEht/NfBi4fP
pLbPQlwd1236vZoKNHDYCIjXbNffm01vcONR1+oMMIII7+48Ia/lr/1gSAKL
1SSdX/LrgJ8k7X1FXXdfOnb5pTyLBnvSPfEknqqNviRo8Qd7h+GpFXHCGGI
dnf0fX7y9yRkvrcvbnf9HmH83KgZYXUYfT/TlPu70w+I68qnD2tSYzzmmWUD
/nge9CwGCDDNAURt1Pa8IQBm0+p6lN/puNmK/bAmkX7fdZokdHzvgF19fr0z
elGTkAl5E01p1JG7025N4s3b3fk4fn/eF/GPtKICWIZ/DLQ0/qECS/xvFP8c
cVP8C7T11RjDWBxrcnjN8kcEWIQghmkCQDmUopj8UxVWlRSLLEJgAtGjUcyI
5Xu2D5GDIehcAsIApxUbdZA3RqzxsWJ1RfjnwKernP6X4V8BUE3hH2t60f+v
pZw+fxL6fnLz8PdoGMEf8vnVFBN2VgJIoJcag3wa5XN7z+MY5oCFy01gKMUM
WrbXi6SIGL8Kn3JNhKX40ecIJ0nZJEcsGe7uPOS3rElfB86+BBTpV8STIB+K
kqLVA0D/Sydn5/y0vheSJueee4k2iur1URL78cVqRfK51d9msdUo80PqdcK
27wNqiJ457L0+8Gv99pc+HHplQRNnYsS0tR5A03eQ0n0GZ4Qo7q7w+ry7k6z
nj0Q1oFgs16dDCzH13iXKzqQnp8/0zh++UzirYLi4KVjUV9CE0qTQ4g/B6kH
Yb9H+jWkSI+fPfiU6fCT5H9vpexfQP9TUJr/kaqW/L9u+S9XcZrVAUVJUm2z
iA44TLN/VKZKIC8pNXc5FhhPSlGZ1QLjU/OaklIB+bDn401Rj2basFTf42W0
yvex4t/xLoGySfuPjpQE/lwBf13BJf63Tv7rVN9DA1QQVjQu0SsQaxkxcBaS
8UgEbQIc5lmy6w0YVwWFDYawqb621zJb6JDqoeH+4f7+wkBbu9XPh1Zlzo+
neGFQ17yRDS0VKy7pgSnzBXhlE3JcLzXL2nANqn/AYxRBv9IK/G/Nf0/npn/
9bz5v+jUPzPzH1ejyX/ZjH8xpL5z0aeJef+80pr6C0/50UgXk76envQnc7vy
iRmHGk3CleWnyv8c+sL+wx1NUXRv0SP5Xy/1/7WU0ewpQQCk7x4tm1JN14Vv
```


BcHYtJM5B6pqNfFKn8dIgwbiXcK2o1EIkIENDCBQ+ITPJ3+bdFEH0mBILZsd
n1n2x0eiTkSAqfiRFk1i4UPhaEdjdSFTjVqDkZUKmnkSgFrIsDM1EEUXF+Yh
aCT55f71JQXTmCsqiE0lvacsN8P/9HW4Wu2vgP1HV9PyH4ZKyf/b5//rvHoF
FwCGNAzSgqCwP94EM4ofrs7z+nFwAk46Xp+2Ha81tmg3/eDkJ23Hbaf9+WJ
E7Ypv9GzWH99JSd0F1841I3qjs8PHvrH3lX2WI6fcGaG0398Jtm+a8mcyGU+
eqKJZJkUnJ5btkWybDitpm9t1P8HVAVn9D+ASvyvU/7D0MSmpi0gcDSGIQ0E
wKRGQELX4qiSATqkzqVA13wJEaQLPYhMg8PZp0msjwYcYZZvo0EHASQ/ejbi
TzDgn9o2YYi2I8MPYayZlFkQNNFK7ruXR1uxwohM4jHbGUl+9/JEv7iiCWDH
s7wuRbkonSPg8cZmJLyPcf5vcsY0Nuz/0TWUnf9L++/22H9nZ27jQy3CpvDz
83980M/BPvBawVXawz5mkHjIzggLQ+oQzhZWFxUwEs+677Mztiz801iGSIXo
2o59k5f3sAsvMAyv1jLc5I0a9Dbr/1FB1v8D9BL/6yjPT745PTs/eR7hNRf8
GfwwEg4C770cv1PLr7hdy147ttk87NovW51uED4ZEho8ejQ2+uafkiSG8Vjm
o0cBR9nnHJMDlMbNThCQuPxdxdQhH4KQ16PWUElS00HA63r+0Itu8r0XDo/6
DG4S/1jTM/iHpfz/qeLfQAafJK+HebgU8ygP8/euBfr7P00fknVwA3qxUfuf
gtQM/lwt9P9uCf6LiFrZo6NALagqHdRBMLMIyNjd0a1Jo/E3I82bptD9bc9F
bhcihBWYmqZHN0vH+U5l/ynU51sroJrHWFG48hxzHX/gqbaUDCKZ6a/xM84e
LxIogxWE8rppQnbpzjJ1nePKjNB0lAwluU6gTE4nIAPE8darD2q+gZAYqbDm
U3q6Mowy5qrdP8v9/xik+R8qpF9/3fYf/mIVdCyotYJBgruSTJWdFyq6ymhF
h3lsF8lXpxXIGeuS8jfehU3Kktcb2WCRkeE9fXfHqUnx+BQRQ4AZmYS7na4H
ke/xP4CMEstCoFgYiouafQTp5jTqFjj5d3ekiZ2qoj3IwWpyyL/nnXHoVTS1
R+uX/M0d8KarFYQrSLl/Q6w5lZTL0ICyXJf/4Ybt/1pW/+dILv1/PfJ/Betc
Vq5gNdYBAtpyWBiq0DpXG9ir/JpXHwVTB5T5g6A5I+zenyzvX1ozon8bItgh
lt2ZCQqLh6Ygfe9i6CMcXEDKINwJhG+w6bkQeYHb9ZiIC5t10D9YugYQArNi
v2EedZMziIkwlpI9krtscCZ4QFyHMCV2CaYcgg8q2teZqjA0D0tXzV4TRRVR
quKN2x94J3sD192s/x9n7X9auf5/y/R/MUzkCgCy+DBPtZ1baWTM00Y5+9Y
sBx4PCSRQHYLPuy27EIs7sXaIj9MLgAxGI2JwVyaU2WBGnkGkL6W6D4L3P7
JQJ1p1DkTTnKnHg/p9a8io9r220v5J0dDAJns+v/Z/Cvxfgv7X/r1v/SBrwv
n/uDkNYr6EGKGyro5EZYAyDMdR8jb03YvFX3EX/MiwsYD+VIWZSF3NDuyKRD
USOQ0TdcANMYyJYZ0qANL+Q29tQpfXCq4ZM/XKw9yr4th20qt7kYUzzib5XG
s1INLMuHfcdjA3vD+h/WlDT/Y1Ta/zBN/100H7FkFLe13Kdg5vgUqtE4W0qK
7mjZ7TV50JzrRDDL3CjL8B/2aNjeqPwHQqB/u1rqf2spt/gAoAFloe015Cd8
JPhwtPx01XTDfHXn7o/ViuNZlFq330YkobwaeAM2IG71xfPTt5+Hvtygt9gg
oF/80mHe7fCuE/7ioBmQN1df7h8cHt6p/dIJ77YJqyv7Q+oE1q07fcJ+MfTf
Wq9C369+bv1UnCY74RdHy8NPxpYdZVH4ZvVWRdPuvLr74+23I3vVvYePa7/9
M+npC/72mfRQ/PW3I6PUdduTa+YK/S71lLc/xh/g7buv4k/oTuVWVY5v1Bt4
FvFSxi+bn/5MhJbcvv327p1XvHKlvvfT3737/bt/f/dfP/3u3e9/+qef/vHd
f7z7z71IYH4VXfWtcvvuj/zKcjUh/I5xPPQDq/LrR/KtKu+B20fizx/ffvH5
/Xu13x7s/dVvvvybv06k0VIGRIYqXWMMpIV1KMJT6Wv06YyT+LeUWDTg967c
ufuKd3/t//7w33/813/749//y//+8x+2dw3Mpy3/XUIVo43a/zRdy8p/Zfzf
uuW/F1z+++qoWAYgy4etYXWex/euggBSVVXh5YhXrTZdhwN/NrC0363eb/se
vS+W+jU5UcmRwWF0jczTHPaIZQV1xtJ29rEF4q6iqWAcx5esNnu7GdKMB7+i
K3recyZiilRtIvjeBE6YD8jqdxPkybtgEG7a/wcy9j+slfk/t0j/K5oF840W
BRmRwR8v9A3Ew3U2Q5TFbKcjsngjaxoLomK8IPyN+T1aLEvofLCPIoNtP+iR
MCvc4Ei6uRchl/eMdK/f7A0ksca4vsent70D/uW9g/HhjF3+ru+1Vh0AvAT/
mqbjjP6nl/7/7cF/pESdSiKISyKSqCd5pEc135a47B/QHv3AN6kf+AIaHfD7
uWt6axrv05l9fKV+SYMmXxv8nVJ3AE9FJ+im4hWrVpVZ23e8vjGq2hS3ivS
OSlX+p7yt++dsP3UfykeQGi0MUeTHnnD1UE3epyYsMXDrvLrBEA0NI1t/B9D
PB6K7UjjXclYqce3SNwhGiM388/4+aNf3aWMTZpw4x8Ss/d4suFSeED511fr

/3c85JyJrIAMETkAJsLCi7hn5Me85k2+ZIG1mjTqePmmIS3Hv+vqrt8fc4Yc
cX99HXeK2pRjA+QMhdCsmnaZrQFna7zM1lBSNfLuA9J1jjN1oJmqc5q9zmxG
MFEEn5zp6qk05zw01dJ2c66ip0nnPg9N1cq6T6u0XUk3K1En18mVenXQ/S1L2
XiB9nWwdxcz8XjmVjGylnFr68p9e0QrUUQvUwQXqoAJ1YIE6BYa0ApbXMZdX
MZZXKdDJBfq4QBcX60ECHVykf5dUyWiVo8Q5Vn1i1xkrmusS9tbxips0ouu1
MbMbkC3fY88091MIynHDVrjqAMC1+X+AmtL/VaWM/1ur/p+fS0tmovxMaCoZ
v2Ey0H8yBpwjHPF75KVcHm+irnCbHg3csEGfD3ETQUMHW0IE7pBZDutvNv97
TvXHuf/XesrN7P+3F0Nu7140D0wz8i2J+x8PRRF6QBpN68jmGuZVnjNOA3NS
cwpayJyVb9gvt/4b43/z8f8o7f9XgVLm/9q0/f9QcoPlFBBD+7VNZZNPYoaB
5DemXSQRMIozs2f8feosJUSjM04GavnUd8TWS1Fw/zDw4k8GwKE3WlQ0cv4p
iIuUGkYFYlWvuamLIDIXCRbK+gWLxYrFtgkWZf1Y+F8QYLhR/U9Haf6HoIz/
XSf/58t/czPAfPBCaBgKnuX40Qi0u94F1/04x2METJNYyuk3GVJ0lCMDKoaJ
DGNJvg9lMryIzHG015Ww/Pabj3pNd9P5f4Cajf8s8z9slf537WQ6IpUCRMVo
whe7NuSZjcRyowRRXEPVbntCBPQDp02GtuUT50NAhT5yAjbdIUhRuTA0VRIZ
wMxooHrI1srs/hPS1+FB3yW0J2B9zh+b3maS50tDciWfViT8obxrh1KPeFeS
3wxpyCQSUImJ3SUytUS1BpXeidXtUD57DuuRsNleJQksWz9ScWb+L+M/t9H+
815xnQirUF20/UtyDHZE5kH22rLdoqncS1PPjeAfbg7/0hf2p/jXI/yrpf1n
LeXpyfdPTs6//e7hdjAAXMIacyX1lKC+mCnmJ/9b7hFG5vuI9HnS/FbYaHoD
N3RWRaEsX/+LU/jHGij3f9uI/ZcDus+F14LrP6C6DHfj8Z6fbTnXHxSvt5pC
fkISo6FqGmKbqKMEkkanmIXPgQtIa2LTFuZpkYR4vg4zPpLRYIrYv6NMWgKd
bbFA1EXGaKs1foaMRf2ajAXn0rHhjSdUK0WyTRav6a7Y+1NA/pvJ/xXPyj3
f964/2+e9Req+EP1wci0g5JWYDE0Z5b1QSUYIQUqhkm0CQCcBcKBUJHEfrNP
pqIPM/TK77Pr10h0MydSCK0tVNDzA0qY7210/x+Y0f+Ucv3/esp4/0/e/9IH
7+05GUyEo1js70mI7LymCaHI4z139/ZyF88Nzv/+JXVZc7Pxf1BNz/9iS9AS
/+u0//qkX21Q2q8dHEXbj2oIAHgZyCBGJtIHZVBHPA7jSB9+eXDBGcYbxoE/
XXeapRVFKYTNbc49WoNhFvH84jIn0cXADxtks/m/dax18V/af7dH/o/W/z8J
qtKvqpLIyxPPo1nR/36hzX9zNr1SknHA4yEZqQGLYvvNtNBEdNcr1aypQAUF
aKFwoAfaxM5XN2AjDmiLDMLN7v8rkV2k8Y9L++9ayor3/zN0BJV2ByIFGot2
tpsMwxkbWbjSnDPm7uknJvypcXK07n88CnT+4LyJIFyTnvrfeSR5w/Zedth
Z822SEHi98nFgMocB6QuAjQyKP9Ziga84xvEaoabxT9K6/+gnP+3Df+L0J+L
fdM0EII1L/xMR18XYZWwwEW21bg8UkBV/KdMFwEu2tAtqVXF4LWReKEFexHP
7CG6eLcd/qR07njwMw0j4b+A2Ddhs/hXtaz9r4z//NjxbyyDfjzwUtBP3Vv
cGufJ3NMjcug/xFs4/PehXGfigSr3QJwqf6PMvEf0izxvyX4z0/ADhcZ2BCG
zLL9+Wb7pK1ggfUfCm0cdtqTQDCKGSbCYgEJVxBNBSB+0EEME3KZjGqIFNC3
zEviuRdo6AWuR4e+3SHMDhZ2QQB3t6T130noJFRYg1lvmtitodFC1HpkMbP
VocGgKaumpqGoKG0H2DGjHvKbW0S/1q0/h9CqAGhKEbyv1r6/9fq/1MBks5o
c0k0qfTCI5fEcUnDpe8F/zdvXlu02LGTDbPpj/iphVkhE1c1ogVKmNwXaWc4
MRDAaLWIgbCuQwQwH0d8ICXwf8wSax/unk8YBb4tgk3bHrzwBz+IPyChQ1tc
KxBZycPgSj62QxrwW2ATAWggg/eUeBd/ivuYBpdzvIEB53gtsSqN39a/pEE4
CKi0d+pJzwK/xSHL9rbT18Cs/sbXf+lqJv4HqqX9b7vs/5GR71VWCZi/djId
G1ogJcDCZADxUB2ZBkVwkMPwNrd0W42aT8crc7y/IZvXU1se9dwCijq4pjw
mUWdW7MHwIc6AhjtORvO/wHUZP5vDZT7f66xpNd/HfJG1gcVDBLgVD4pzIW6
H7TiaAAEMVwUDZaEl7n+AGM6s4/Bn4Lg/swX/W6T6XLP6dF9IUbHJQoN560N
D3r0a2rtJ75gTsvj3yTq8gmctCg/wK+c+MM0SGsRpWz/ejQmemWz6z+wosFI
/tcVCHQ9Wv+taqX+v5by50Ts7PibWADoDnp015+P9nv5lWZ1dtsd2PZVji7P
ZXSG6I08uEA9tVfn2e5ly9BVjGQuDK0MIdFsLuar0JYVeXzoMBB7ByTIQ8ew
pTVsFdGGZetY0foB+oMfoPvmB2gMmtAdkr/8i0Gj93K6taU06gcu9Cf2IJo8
WGbH71ErZ5/8/rzUp5GkEiHrIrIj17TtyJloeF4JLhKCxNpGjps+GLru0Cq

ji3d1hA1NGJqTaQphz10cG6rztpE0ewHfug3fbeeJT5xHSIUkt2dh0Q07nxA
93mvS8f9QIJ8/pUuvYZATcXSN0/Od3dEiId830KPPF10Tq8vePsgEkikz5uD
gFH2RZ5gpOmC1mQ5/aCL1tEX6B+x1N5hUpswiUhia3jJ8SSuGkpC1KsWuWH+
ZHAYb78gPlb70pue9NBhfZ85sb0YhPx3b/f494eRMMcfrR7Qi4ET8N/Gdlw6
7yoLWwR+9yf/86d/fucfvv3/9q4GuInjCuvPNjG1XUckULCZjWljYi0ydyfZ
khyBgUnqmmLXlh0TQ0rOupN1knyS705WDQQsU9xpQobEU0qnKcEzbQdCUmh+
GDJDgaaTZkhpXZuGwkyaMJ30dCYwwxTy1/xQunsn25Is2U7byCHdz5at3bs9
nfbe233v7XtvBz4o00cZhvpNo6jqDwa9npOLb8nNqSw0Ghbm60DciZP0Q/2G
s7DFMIxe1UNGG95gmKPfo+s2MHozLBg/T5+DmoXVqxhbTb1mQ6ubMsNiXJhj
xjwieENIW2WpYliEK/PMc9ysC0pDMk8Vwi/hqnxzDiJyQaHuguW4PNcM3LgM
WnhZAWuQdit4cXfyIL6wrVTCxfMLoA2N5Ug8pRyMvR0V7ahIxYuw439zR4vg
V3DZ00/WyQMNj01P7Ad9js4Ye0wHY983xGK6vY8+d/DwqgfawB1o3xt3PLLt
lXeKmNPrGzZsyV35x2e0795zqmRlj2vngaKSt93KcF/JgXXMyku95pZHL/yu
wXePac/7t26p2pC79RHTLXdKF687//QPB1iXN903UPbik64/HRp+b7jq4pGm
6yd9hz6+eNSgVLV86+K87S1rj147WTP88d8GnzjGXTY9Tr9T7fDh2ET+gKl
xaZ6WDdwe/oRc+A2oStd9fwwNkym+eK4Y0vNJg0afkrx20LT7aZ54tBd0pEt
733n2ncPyh/s3ffCLx649HgKnRhRx3z5sqfsics7H/r9mdM3nnqpb1HfjgPP
fg0c6f3ZzmVv7x06FSuvf0zEt58/I+39UeX7IXCmbf2FjT95fDXffee+e5d8
1Hk6cL7yw/oSK2hp3j/4Qnj0t+0LI+/+av7qmsrj7uGLP750LmdwU/WA8vMT
z/K7rtddFV3bdFTD7/VXmz767/Kaw64By8epvr1f4b9+r0I90HW7BNUAg9N
sN1Q7Dm4YLynbjFSiWwISya05FBFE3TPQBvTDpdMHDVR6DEUNi15lXt9W+DJ
k3fCN+84+w2+r/iXsHnipCLqXrgG5uf00SqMebBw/L3BYND1oWc2UdbXwjzt
NvOS65emkL4p1rfgyiHbP/NubN3+w2Nf15Z0vrZJ+uT8qlppUdUpz47IuTeu
rnj3H/ndf9m2YkAsbXp57Wslf3Njh3Dj/mtXvcwbweGny5nWg+e+17YAND5z
5MFjdfnd0x96af/ipR8u4KjjlV9tfv3BT3y/vrL6zb5dYduGtS/v7m1564J4
Nv96W3m7v/Sjw3/Pe6U/3chssegIPkdQEGN81gGA067/J+Z/0vR/pAsQ+T/L
9j82yIq1f1Y0dQnx80ZGgMSf1FpN1pfYYNiXrPSvY9dbvI1SG9vMuZGGC6sS
HfxU0ktS5hOX+tf3bt7ca6HRaVjQQ79YuhMjXR28hFqKQPMQ1s5in002PLUi
aaE03QXPaqZ4bygE0tCjRY8N33UohAuS+i+N7GqfqU1P0fZJdqmb218A/sdC
Klaf5FnM/1JVncL/NhvX/5kd+99/lv/7v8r7DRJoMCD4fVY2wIlstz8Q5UV0
ZmUvNvx1sPPNKCP49GkgUxq517SgVvRUrYKs70d83tT10W+6gcJM0Y522CHS
5vE3Sr7NhrUNjW0NQLF00ZiV0ehleDHNouyU7k+QdQicV+T5NKZHYrKsz3/
22Aq/1sZSPg/6/wv8V0hhbeg/uhMb/QHIJM/gEp9GSgeL9VBR6IvUMYPSjYl
BkNoSLEglTcUkTx8uoHFYaUd005IldMwPYcET4Sx0mx2HELIE3EEoU+iGatN
mnYgoRm7IzG4MLNB74sRMRgRA7Kn67MdAaaV/61MKv9XEf7PLv+nrJtrMTFO
JaTgTT010BiPrvjZiAa0U6VIOoGzwf9d9CzzP00n8r+NJvp/VjAz/39B7hCd
SHZlqh20vZoZn6V9ioITBkSj0eXpxH7EmKlcrdKaqv77esM+XmQVnrOkpnGd
mIytDpjo4Dt5Ns6ccI6aSSSTQPXjxxlkf8omb3F2C4lvx/5YtNiLOev4fCm/2
o63/U9U2GmrPyT/c1YwJtnTEAIxoCsQGfABbBUVNBg5MXISjMZG+9G/kRdH
jo48P7ob16VUjfaP7gIWMHJqtG/s2MjRqXxiHPYUnqQY2uqgkweKOFHircAC
eNIPMH6al6I855VXuadIJZQxGImit02GMkgTPBfRvBRo7J48eWxbNqOUA0nc
+B32SRmGPL2aAic9Y54Af0jzt6UowU2EqCyIPbPr/40H/VT5D4eEkvE/u+s/
PUIPH+is9fhyQbJwouRJUppQ8IaUQxH1AZsoHDYcz0H+9M1BDQAADbDYmdKF
BjDKjn8YRizI90p4pyHgBGX1oDkk82IHL3WCjRs3lpXfVs+1bD0cha/WToqJ
wn/wlV0AJ/n1c/ZpVqI0ik+zCJU86qu3pPp6e2Q18ySEDvxRCX7k8W3lAbgb
nax2B8BTneY20i6Hjs12jAPKcfftXp3WwEfvy8X0IM/V8SyHQ0wAEPmoV6sD
8b20Ada6BTGCaiYfbNWeida+DYmvrWJEjrDB+/FhJ6ipWbYmVwqmnbs0nzf0
YwPyPy1oYkJz0+IPSGAVE0ee41BsYq7GIIYtagR5Ijx0M9YlqMLwbYB0cQw4L

eLpLoor449N6CFMGAHGhwCEE/HYAsA/ueBdrV0RvsTyhdjBVZV9022zL0ey0
rxW/kAs1GotExBE6aAJ0ljVFQkjVALIiCSIirzJMXhkJC10K9bZG664xN2BU
o35L7Z5lbA/htFNqQDe6d7ic+fRZgDLmALVmZ2rfzCP2886y/k/DyfY/ov/f
R0t/U1n60n0yZ6USBuA4yfmXKS/oFbvoEB20MpItAjVzX5Jpj6GpKVb2ZrTH
G7YiCKm2A0hS/xEQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQ
EBAQENyk+Dc2+ZBPABgBAA==
-- END MESSAGE ARCHIVE --

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Copyright (C) The Internet Society (2005). This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

Acknowledgment

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

