

Groupe de travail Réseau
Request for Comments : 4475
 Catégorie : Information
 Traduction Claude Brière de L'Isle

R. Sparks, éd., Estacado Systems
 A. Hawrylyshen, Ditech Networks
 A. Johnston, Avaya
 J. Rosenberg, Cisco Systems
 H. Schulzrinne, Columbia University
 mai 2006

Messages d'essais de résistance du protocole d'initialisation de session (SIP)

Statut du présent mémoire

Le présent mémoire apporte des informations pour la communauté de l'Internet. Le présent mémoire ne spécifie aucune sorte de norme de l'Internet. La distribution du présent mémoire n'est soumise à aucune restriction.

Notice de Copyright

Copyright (C) The Internet Society (2006).

Résumé

Le présent document d'information donne des exemples de messages d'essais du protocole d'initialisation de session (SIP, *Session Initiation Protocol*) destinés à entraîner et "torturer" une mise en œuvre de SIP.

Table des matières

1. Généralités.....	1
2. Conventions du document.....	2
2.1 Représentation de lignes longues.....	2
2.2 Représentation de caractères non imprimables.....	2
2.3 Représentation de longues chaîne répétées.....	3
3. Messages d'essai SIP.....	3
3.1 Essais d'analyseur (syntaxe).....	3
3.2 Sémantique de la couche de transaction.....	21
3.3 Sémantique de la couche d'application.....	21
3.4 Rétro compatibilité.....	27
4. Considérations sur la sécurité.....	28
5. Remerciements.....	28
6. Références pour information.....	28
Appendice A. Archive au bit près de chaque message d'essai.....	29
A.1 Messages de référence codés.....	29
Adresse des auteurs.....	31
Déclaration complète de droits de reproduction.....	31

1. Généralités

Le présent document est informatif et N'EST NORMATIF sur AUCUN aspect de SIP.

Le présent document contient des messages d'essais fondés sur la version actuelle (2.0) du protocole d'initialisation de session telle que définie dans la [RFC3261]. Certains messages concernent l'utilisation par SIP du protocole de description de session (SDP, *Session Description Protocol*) décrit par la [RFC3264].

Ces messages ont été développés et précisés aux événements d'essais d'interopérabilité de SIPit.

Les messages d'essais sont organisés en plusieurs sections. Certains visent seulement un analyseur SIP, et d'autres visent à la fois l'analyseur et l'application au dessus de lui. Certains messages sont valides, et d'autres ne le sont pas. Chaque exemple précise clairement ce qui rend un message invalide incorrect.

Le présent document ne tente pas de cataloguer toutes les façons de rendre invalide un message, ni d'être exhaustif dans

l'exploration des messages non usuels, mais valides. Il essaye plutôt de se concentrer sur les domaines où ils ont causé des problèmes d'interopérabilité ou qui ont des caractéristiques particulièrement défavorables si ils sont traités de façon inappropriée. Ce document est le germe d'un plan d'essais, pas un plan d'essais en lui-même.

Les messages sont présentés dans le texte en utilisant un ensemble de conventions de balisage pour éviter des ambiguïtés et satisfaire aux exigences de présentation des projets Internet. Pour résoudre toute ambiguïté restante, une version précise au bit près de chaque message est encapsulée en appendice.

2. Conventions du document

Le présent document contient de nombreux exemples de messages SIP. Bien que SIP soit un protocole fondé sur le texte, beaucoup de ces exemples ne peuvent pas être rendus sans ambiguïté sans un marquage supplémentaire à cause des contraintes qui pèsent sur le formatage des RFC. Le présent document définit et utilise le marquage défini dans cette section pour lever ces ambiguïtés. Ce marquage utilise les conventions d'étiquette de début et de fin de XML mais ne définit aucun type de document XML.

L'appendice contient un formulaire codé en binaire de tous les messages et l'algorithme nécessaire pour les décoder en fichiers.

2.1 Représentation de lignes longues

Plusieurs de ces exemples contiennent des lignes non repliées de plus de 72 caractères. Elles sont insérées entre des étiquettes `<allOneLine/>` (*sur une ligne*). La ligne seule non repliée est reconstruite en enchaînant directement toutes les lignes qui apparaissent entre les étiquettes (en éliminant tout saut à la ligne ou retour chariot). Il n'y a pas d'espaces blanches à la fin des lignes. Toute espace apparaissant à un point de saut à la ligne va apparaître au début de la ligne.

Ce qui suit représente la même chaîne de bits :

Nom-d'en-tête: première valeur, deuxièmevaleurvraimenttrèslongue, troisième valeur

```
<allOneLine>
  Nom-d'en-tête: première valeur,
  deuxième valeurvraimenttrèslongue
  , troisième valeur
</allOneLine>
```

```
<allOneLine>
  Nom-d'en-tête: première valeur,
  deuxième
  valeur
  vraimenttrèslongue,
  troisième valeur
</allOneLine>
```

Noter que ceci N'est PAS un retour à la ligne d'en-tête SIP, où les différentes chaînes de bits ont une signification équivalente.

2.2 Représentation de caractères non imprimables

Plusieurs exemples contiennent des corps de message binaires ou des valeurs de champ d'en-tête contenant des caractères codés en UTF-8 qui ne sont pas dans la gamme ascii. Ceci est rendu ici par une paire de chiffres hexadécimaux par octet entre des étiquettes `<hex/>`. Ce rendu s'applique même à l'intérieur des chaînes entre guillemets.

Ce qui suit représente la même chaîne de bits :

Nom d'en-tête : valeur une
 Nom d'en-tête : valeur<hex>206F6E</hex>e

Ce qui suit est un champ d'en-tête Subject qui contient le symbole Euro :

Subject: <hex>E282AC</hex>

2.3 Représentation de longues chaîne répétées

Plusieurs exemples contiennent de très longues valeurs de données créées avec des chaînes de bits répétées. Elles vont être rendues ici en utilisant <repeat count=un_entier>valeur</repeat>. Comme avec <hex>, ce rendu s'applique même à l'intérieur de chaînes entre guillemets.

Par exemple, la valeur "abcabcabc" peut être rendue par <repeat count=3>abc</repeat>. Un nom d'affichage de "1000000 bouteilles de bière" pourrait être rendu par

To: "1<repeat count=6><hex>30</hex></repeat> bouteilles de bière" <sip:bière.exemple.com>

Un champ d'en-tête Max-Forwards avec une valeur de one google va être rendue ici par

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

3. Messages d'essai SIP

3.1 Essais d'analyseur (syntaxe)

3.1.1 Messages valides

3.1.1.1 Un court INVITE tortueux

Ce court message, relativement lisible par l'homme contient :

- o des sauts à la ligne tout au long.
- o des caractères avec échappement au sein des guillemets.
- o un sujet vide.
- o des espaces LWS entre les deux points, les points virgules, les valeurs de champ d'en-tête, et autres champs.
- o des valeurs de champ d'en-tête séparées par des virgules et des listes séparées.
- o un mélange de formes courtes et longues du même nom de champ d'en-tête.
- o un paramètre inconnu d'URI de demande.
- o des champs d'en-tête inconnus.
- o un champ d'en-tête inconnu avec une valeur qui serait syntaxiquement invalide si elle était définie en termes de paramètre générique.
- o un ordre inhabituel des champs d'en-tête.
- o une casse de caractère inhabituelle de nom de champ d'en-tête.
- o des paramètres inconnus d'un champ d'en-tête connu.
- o un paramètre uri sans valeur.
- o un paramètre d'en-tête sans valeur.
- o des champs d'entier (Max-Forwards et CSeq) avec des zéros en tête.

Tous les éléments devraient traiter cela comme des demandes bien formées.

Le champ d'en-tête UnknownHeaderWithUnusualValue (*en-tête inconnu avec valeur inhabituelle*) mérite une attention particulière. Si ce champ d'en-tête était défini en termes de valeurs séparées par des virgules avec des paramètres séparés par des points-virgules (comme le feraient de nombreux champs d'en-tête définis existants) ceci serait invalide. Cependant, comme l'élément receveur ne connaît pas la définition de la syntaxe pour ce champ, il doit l'analyser comme une valeur d'en-tête. Les mandataires transmettraient ce champ d'en-tête inchangé. Les points d'extrémité ignoreraient ce champ d'en-tête.

Détails du message : wsinv

INVITE sip:vivekg@chair-dnrc.exemple.com;unknownparam SIP/2.0

TO :

sip:vivekg@chair-dnrc.exemple.com ; tag = 1918181833n

from : "J Rosenberg \\\\" <sip:jdrosen@exemple.com>

```

;
tag = 98asjd8
MaX-fOrWaRdS: 0068
Call-ID: wsinv.ndaksdj@192.0.2.1
Content-Length : 150
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.exemple.com;lr;unknownwith=value;inconnu-no-value>
v: SIP /2.0 /TCP spindle.exemple.com ;
branch = z9hG4bK9ikj8 ,
SIP / 2.0 /UDP 192.168.255.111 ; branch=
z9hG4bK30239
m:"Quoted string \"\" <sip:jdrosen@exemple.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 49217 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC

```

3.1.1.2 Large gamme de caractères valides

Ce message met en œuvre dans plusieurs éléments syntaxiques clés une plus large gamme de caractères que les mises en œuvre n'en voient généralement. En particulier, on notera que :

- o La méthode contient des caractères non alphabétiques provenant du jeton. On note que % n'est pas un caractère d'échappement pour ce champ. Une méthode de IN%56ITE est une méthode inconnue. Ce n'est pas la même qu'une méthode de INVITE.
- o L'URI de demande contient des caractères non usuels, mais légaux.
- o Un paramètre de branche contient tous les caractères non alphanumériques provenant du jeton.
- o La chaîne entre guillemets de la valeur du champ d'en-tête To contient des expansions quoted-pair, incluant un caractère quoted NULL.
- o La partie nom de name-addr dans la valeur du champ d'en-tête From contient plusieurs jetons (au lieu d'une chaîne entre guillemets) avec tous les caractères non alphanumériques provenant de la règle de production de jeton. Cette valeur a aussi un paramètre d'en-tête inconnu dont le nom contient les caractères de jeton non alphanumériques et dont la valeur est une chaîne codée en UTF-8 de gamme non ascii. Le paramètre d'étiquette sur cette valeur contient des caractères de jeton non alphanumériques.
- o La valeur du champ d'en-tête Call-ID contient les caractères non alphanumériques du mot. Noter que dans cette production :
 - * % n'est pas un caractère d'échappement. Il est seulement un caractère d'échappement dans les productions qui correspondent à la règle "escaped".

- * " ne commence pas une chaîne entre guillemets. Ni ', ` ni " n'impliquent qu'il va y avoir un symbole correspondant plus loin dans la chaîne.
 - * Les caractères []{}()<> n'ont aucune signification de groupement. Il n'est pas exigé qu'ils apparaissent par paires.
- o Il y a un champ d'en-tête inconnu (correspondant à un en-tête d'extension) avec des caractères de jeton non alphanumériques dans son nom et une valeur UTF8-NONASCII.

Si cet URI inhabituel avait été défini chez un mandataire, il aurait transmis cette demande normalement. Autrement, un mandataire va générer un code 404. Les points d'extrémité vont générer un code 501 avec la liste des méthodes qu'ils comprennent dans un champ d'en-tête Allow.

Détails du message : 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)
@exemple.com SIP/2.0
</allOneLine>
Via: SIP/2.0/TCP host1.exemple.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?./;.*
@exemple.com>
</allOneLine>
<allOneLine>
From: token1~` token2'+_ token3*%!.- <sip:mundane@exemple.com>
;fromParam"~+* !.-%=
"<hex>D180D0B0D0B1D0BED182D0B0D18ED189D0B8D0B9</hex>"
;tag=_token~1'+`*%!.-
</allOneLine>
Call-ID: intmeth.word%ZK-!.*_+@word`~)(><:\v"[]?}{
CSeq: 139122385 !interesting-Method0123456789_ *+`.%indeed'~
Max-Forwards: 255
<allOneLine>
extensionHeader-!.%*+_`~:
<hex>EFBBBF5A4A7E5819CE99BBB</hex>
</allOneLine>
Content-Length: 0
```

3.1.1.3 Utilisation valide du mécanisme d'échappement %

Cet INVITE teste le mécanisme d'échappement % HEX HEX en plusieurs endroits. La demande est syntaxiquement valide. Les caractéristiques intéressantes sont :

- o L'URI de demande a sips:user@exemple.com incorporé dans sa partie utilisateur. Ce que cela peut vouloir dire pour exemple.net sort du domaine d'application du présent document.
- o Les URI From et To ont des caractères avec échappement dans leur partie utilisateur.
- o L'URI Contact a des caractères avec échappement dans les paramètres d'URI. Noter que le "nom" uri-parameter a une valeur de "value%41", qui n'est PAS équivalente à "valueA". Selon la [RFC3986], le non échappement de composants URI n'est jamais effectué de façon récurrente.

Un analyseur doit accepter cela comme un message bien formé. L'application qui utilise le message doit traiter les expansions % HEX HEX comme équivalentes au caractère qui est codé. L'application ne doit pas essayer d'interpréter % comme un caractère d'échappement dans les endroits où % HEX HEX ("échappé" dans la grammaire) n'est pas une partie valide de la construction. Dans la [RFC3261], "échappé" ne se produit que dans les expansions de SIP-URI, SIPS-URI, et Reason-Phrase.

Détails du message : esc01

```
INVITE sip:sips%3Auser%40exemple.com@exemple.net SIP/2.0
To: sip:%75se%72@exemple.com
From: <sip:I%20have%20spaces@exemple.net>;tag=938
Max-Forwards: 87
i: esc01.239409asdfakjkn23onasd0-3234
CSeq: 234234 INVITE
Via: SIP/2.0/UDP host5.exemple.net;branch=z9hG4bKkdjuw
C: application/sdp
Contact: <sip:cal%6Cer@host5.exemple.net;%6C%72;n%61me=v%61lue%25%34%31>
Content-Length: 150
```

```
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 49217 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtptime:31 LPC
```

3.1.1.4 Nuls échappés dans les URI

Cette demande register contient plusieurs URI avec des nuls dans la partie utilisateur. Le message est bien formé - les analyseurs doivent accepter ce message. Les mises en œuvre doivent faire très attention quand elles n'échappent pas l'adresse d'enregistrement (AOR, *Address-of-Record*) dans cette demande afin de ne pas raccourcir prématurément le nom d'utilisateur. Cette demande enregistre deux URI Contact distincts.

Détails du message : escnull

```
REGISTER sip:exemple.com SIP/2.0
To: sip:null-%00-null@exemple.com
From: sip:null-%00-null@exemple.com;tag=839923423
Max-Forwards: 70
Call-ID: escnull.39203ndfvkjdasfkq3w4otr0adsfdfnvd
CSeq: 14398234 REGISTER
Via: SIP/2.0/UDP host5.exemple.com;branch=z9hG4bKkdjuw
Contact: <sip:%00@host5.exemple.com>
Contact: <sip:%00%00@host5.exemple.com>
L:0
```

3.1.1.5 Utilisation de % quand ce n'est pas un échappement

Dans la plupart des endroits où % peut apparaître dans un message SIP, il n'est pas un caractère d'échappement. Cela peut surprendre l'utilisateur non averti. Les demandes bien formées suivantes ont cette propriété :

- o La méthode demande est inconnue. Elle n'est PAS équivalente à REGISTER.
- o La portion nom d'affichage des champs d'en-tête To et From est "%Z%45". Noter que ce n'est pas la même chose que %ZE.
- o Ce message a deux valeurs de champ d'en-tête Contact, pas trois. <sip:alias2@host2.exemple.com> est une valeur de champ d'en-tête C%6Fntact.

Un analyseur devrait accepter ce message comme bien formé. Un mandataire va transmettre ou rejeter le message selon ce que l'URI de demande signifie pour lui. Un point d'extrémité va rejeter ce message avec un code 501.

Détails du message : esc02

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

3.1.1.6 Message sans LWS entre le nom d'affichage et <

Cette demande OPTIONS n'est pas valide selon la grammaire de la RFC 3261 car il n'y a pas de LWS entre le jeton dans le nom d'affichage et < dans la valeur du champ d'en-tête From. Ceci a été identifié comme une erreur de la spécification qui sera corrigée lors de la révision de la RFC 3261. Les éléments devraient accepter cette demande comme bien formée.

Détails du message : lwsdisp

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

3.1.1.7 Longues valeurs dans les champs d'en-tête

Cette demande bien formée contient des champs d'en-tête avec de nombreuses valeurs et des valeurs qui sont très longues.

Les caractéristiques incluent ce qui suit :

- o Le champ d'en-tête To a un long nom d'affichage, et de longs noms et valeurs de paramètre d'URI.
- o Le champ d'en-tête From a de longs noms et valeurs de paramètres d'en-tête, en particulier, une très longue étiquette.
- o Le Call-ID est un long jeton.

Détails du message : longreq

```
INVITE sip:user@exemple.com SIP/2.0
<allOneLine>
To: "J'ai un nom d'utilisateur de
<repeat count=10>extreme</repeat> proportion"
<sip:user@exemple.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>@exemple.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>
Inconnu-<repeat count=20>Long</repeat>-Name:
inconnu-<repeat count=20>long</repeat>-value;
inconnu-<repeat count=20>long</repeat>-parameter-name =
inconnu-<repeat count=20>long</repeat>-parameter-value
</allOneLine>
Via: SIP/2.0/TCP sip33.exemple.com
v: SIP/2.0/TCP sip32.exemple.com
V: SIP/2.0/TCP sip31.exemple.com
Via: SIP/2.0/TCP sip30.exemple.com
ViA: SIP/2.0/TCP sip29.exemple.com
ViA: SIP/2.0/TCP sip28.exemple.com
VIA: SIP/2.0/TCP sip27.exemple.com
via: SIP/2.0/TCP sip26.exemple.com
viA: SIP/2.0/TCP sip25.exemple.com
vIa: SIP/2.0/TCP sip24.exemple.com
vIA: SIP/2.0/TCP sip23.exemple.com
V : SIP/2.0/TCP sip22.exemple.com
v : SIP/2.0/TCP sip21.exemple.com
V : SIP/2.0/TCP sip20.exemple.com
v : SIP/2.0/TCP sip19.exemple.com
Via : SIP/2.0/TCP sip18.exemple.com
Via : SIP/2.0/TCP sip17.exemple.com
Via: SIP/2.0/TCP sip16.exemple.com
Via: SIP/2.0/TCP sip15.exemple.com
Via: SIP/2.0/TCP sip14.exemple.com
Via: SIP/2.0/TCP sip13.exemple.com
Via: SIP/2.0/TCP sip12.exemple.com
Via: SIP/2.0/TCP sip11.exemple.com
Via: SIP/2.0/TCP sip10.exemple.com
Via: SIP/2.0/TCP sip9.exemple.com
Via: SIP/2.0/TCP sip8.exemple.com
Via: SIP/2.0/TCP sip7.exemple.com
Via: SIP/2.0/TCP sip6.exemple.com
Via: SIP/2.0/TCP sip5.exemple.com
Via: SIP/2.0/TCP sip4.exemple.com
Via: SIP/2.0/TCP sip3.exemple.com
Via: SIP/2.0/TCP sip2.exemple.com
Via: SIP/2.0/TCP sip1.exemple.com
<allOneLine>
Via: SIP/2.0/TCP
host.exemple.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.exemple.net>
</allOneLine>
Content-Type: application/sdp
l: 150

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 49217 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC

```


3.1.1.8 Octets en queue supplémentaires dans un datagramme UDP

Ce message contient une seule demande SIP REGISTER, qui est arrivée ostensiblement sur UDP dans un seul datagramme. Le paquet contient des octets supplémentaires après le corps (qui dans ce cas a une longueur de zéro). Les octets supplémentaires se trouvent ressembler à une demande SIP INVITE, mais (selon le paragraphe 18.3 de la [RFC3261]) ils sont juste du bruit parasite qui doit être ignoré.

Un élément SIP qui reçoit ce datagramme va traiter normalement la demande REGISTER et ignorer les bits supplémentaires qui ressemblent à une demande INVITE. Si l'élément est un mandataire qui choisit de transmettre le REGISTER, les octets INVITE ne vont pas apparaître dans la demande transmise.

Détails du message : dbreq

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

```
INVITE sip:joe@exemple.com SIP/2.0
t: sip:joe@exemple.com
From: sip:caller@exemple.net;tag=141334
Max-Forwards: 8
Call-ID: dbreq.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: 150
```

```
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 49217 RTP/AVP 0 12
m =video 3227 RTP/AVP 31
a=rtptime:31 LPC
```

3.1.1.9 Paramètres séparés par des points-virgules dans la partie utilisateur d'un URI

Cette demande a un paramètre séparé par des points-virgules contenus dans la partie "utilisateur" du Request-URI (dont la valeur contient un symbole @ échappé). Les éléments receveurs vont accepter cela comme un message bien formé. Le Request-URI va s'analyser comme si la partie utilisateur était "user;par=u@exemple.net".

Détails du message : semiuri

```
OPTIONS sip:user;par=u%40exemple.net@exemple.com SIP/2.0
To: sip:j_user@exemple.com
From: sip:caller@exemple.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 Types divers et inconnus de transport

Cette demande contient des valeurs de champ d'en-tête Via avec tous les types de transport connus et teste le mécanisme d'extension de transport. Les analyseurs doivent accepter ce message comme bien formé. Les éléments qui reçoivent ce message vont le traiter exactement comme si la seconde valeur de champ d'en-tête et les suivantes spécifiaient UDP (ou un autre transport).

Détails du message : transports

```
OPTIONS sip:user@exemple.com SIP/2.0
To: sip:user@exemple.com
From: <sip:caller@exemple.com>;tag=323
Max-Forwards: 70
Call-ID: transports.kjh4akdnaqjkwendsasfdj
Accept: application/sdp
CSeq: 60 OPTIONS
Via: SIP/2.0/UDP t1.exemple.com;branch=z9hG4bKkdjuw
Via: SIP/2.0/SCTP t2.exemple.com;branch=z9hG4bKklasjdhf
Via: SIP/2.0/TLS t3.exemple.com;branch=z9hG4bK2980unddj
Via: SIP/2.0/UNKNOWN t4.exemple.com;branch=z9hG4bKasd0f3en
Via: SIP/2.0/TCP t5.exemple.com;branch=z9hG4bK0a9idfnee
l: 0
```

3.1.1.11 Message MIME multi parties

Cette demande MESSAGE contient deux parties de corps. La seconde partie est codée en binaire et contient des caractères nuls (0x00). Les receveurs doivent veiller à tramer le message reçu correctement.

Les analyseurs doivent accepter ce message comme bien formé, même si l'application au dessus de l'analyseur ne prend pas en charge multipart/signed.

Des exemples supplémentaires de message multipart/mime, en particulier des messages S/MIME, sont disponibles dans le document d'exemple de flux d'appel de sécurité [RFC6216].

Détails du message : mpart01

```
MESSAGE sip:kumiko@exemple.org SIP/2.0
<allOneLine>
Via: SIP/2.0/UDP 127.0.0.1:5070
;branch=z9hG4bK-d87543-4dade06d0bdb11ee-1--d87543-;rport
</allOneLine>
Max-Forwards: 70
Route: <sip:127.0.0.1:5080>
<allOneLine>
Identity:
r5mwreLuyDRYBi/0TiPwEsY3rEVsk/G2WxhgTV1PF7hHuLIK0YWVKZhKv9Mj8UeXqkMVbnVq37CD+813gvYjcBU
aZngQmXc9WNZSDNGCzA+fW19MEUHWIZo1CeJebdY/XlgKeTa0Olvq0rt70Q5jiSfbqMJmQFteeivUhkMWYUA=
</allOneLine>
Contact: <sip:fluffy@127.0.0.1:5070>
To: <sip:kumiko@exemple.org>
From: <sip:fluffy@exemple.com>;tag=2fb0dcc9
Call-ID: 3d9485ad0c49859b@Zmx1ZmZ5LW1hYy0xNi5sb2NhBA..
CSeq: 1 MESSAGE
Content-Transfer-Encoding: binary
Content-Type: multipart/mixed;boundary=7a9cbec02ceef655
Date: Sat, 15 Oct 2005 04:44:56 GMT
User-Agent: SIPimp.org/0.2.5 (curses)
Content-Length: 553
```

```
--7a9cbec02ceef655
Content-Type: text/plain
Content-Transfer-Encoding: binary
```

```
Hello
--7a9cbec02ceef655
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
```

```
<hex>
3082015206092A864886F70D010702A08201433082013F0201013109300706052B0E03021A300B06092A864886F70D0
10701318201203082011C020101307C3070310B3009060355040613025553311330110603550408130A43616C69666F72
6E69613111300F0603550407130853616E204A6F7365310E300C060355040A1305736970697431293027060355040B13
205369706974205465737420436572746966696361746520417574686F7269747902080195007102330113300706052B0E
03021A300D06092A864886F70D01010105000481808EF466F948F0522DD2E5978E9D95AAE9F2FE15A06659716292
E8DA2AA8D8350A68CEFFAE3CBD2BFF1675DDD5648E593DD64728F26220F7E941749E330D9A15EDABDB93D1
0C42102E7B7289D29CC0C9AE2EFBC7C0CFF9172F3B027E4FC027E1546DE4B6AA3ABB3E66CCCB5DD6C64B83
83149CB8E6FF182D944FE57B65BC99D005
</hex>
--7a9cbec02ceef655--
```

3.1.1.12 Phrase de cause inhabituelle

Cette réponse 200 contient une phrase de cause autre que "OK". La phrase de cause est destinée à la consommation humaine et peut contenir toute chaîne produite par

Reason-Phrase = *(réservé /non réservé / échappé / UTF8-NONASCII / UTF8-CONT / SP / HTAB)

Cette réponse particulière contient des caractères non réservés et non ascii UTF-8. Cette réponse est bien formée. Un analyseur doit accepter ce message.

Détails du message : 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@exemple.com;tag=11141343
To: sip:user@exemple.edu;tag=2229
Content-Length: 154
Content-Type: application/sdp
Contact: <sip:user@host198.exemple.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 49217 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtptime:31 LPC
```

3.1.1.13 Phrase de cause vide

Cette réponse bien formée ne contient pas de phrase de cause. Un analyseur doit accepter ce message. Le caractère espace après le code de cause est exigé. Si il était absent, ce message pourrait être rejeté comme invalide (un receveur libéral l'accepterait de toutes façons).

Détails du message : 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@exemple.com>;tag=39ansfi3
To: <sip:user@exemple.edu>;tag=902jndnke3
Content-Length: 0
Contact: <sip:user@host105.exemple.com>
```

3.1.2 Messages invalides

Ce paragraphe contient plusieurs messages invalides reflétant des erreurs vues dans des événements d'interopérabilité et explore d'importantes conditions limites qui peuvent être déduites de messages mal formées. Ce paragraphe n'essaye pas d'être une liste complète de tous les types de messages invalides.

3.1.2.1 Séparateurs étranges dede champs d'en-tête

Le champ d'en-tête Via de cette demande contient des points-virgules et virgules supplémentaires sans paramètres ni valeurs. Le champ d'en-tête Contact contient des points-virgules supplémentaires sans paramètres. Ce message est syntaxiquement invalide.

Un élément qui reçoit cette demande devrait répondre avec une erreur 400 Mauvaise demande.

Détails du message : badinv01

```
INVITE sip:user@exemple.com SIP/2.0
To: sip:j.user@exemple.com
From: sip:caller@exemple.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@exemple.org>;,;
Content-Length: 152
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 49217 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.2.2 Longueur de contenu supérieure au message

C'est un message de demande avec une longueur de contenu plus grande que la longueur réelle du corps.

Lorsque il est envoyé sur UDP (comme l'a ostensiblement été ce message) l'élément receveur devrait répondre par une erreur 400 Mauvaise demande. Si ce message est arrivé sur un transport fondé sur le flux, comme TCP, il n'y a pas grand

chose que le receveur pourrait faire sauf attendre plus de données sur le flux et clore la connexion si rien n'arrive dans un délai raisonnable.

Détails du message : clerr

```
INVITE sip:user@exemple.com SIP/2.0
Max-Forwards: 80
To: sip:j.user@exemple.com
From: sip:caller@exemple.net;tag=93942939o2
Contact: <sip:caller@hungry.exemple.net>
Call-ID: clerr.0ha0isndaksdjweiafasdk3
CSeq: 8 INVITE
Via: SIP/2.0/UDP host5.exemple.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 49217 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.2.32 Longueur de contenu négative

Cette demande a une valeur négative pour la longueur du contenu.

Un élément qui reçoit ce message devrait répondre par une erreur. Cette demande apparaît sur UDP, de sorte que le reste du datagramme peut être simplement éliminé. Si une demande comme celle là arrive sur TCP, l'erreur de tramage n'est pas récupérable, et la connexion devrait être close. Le même comportement est approprié pour les messages qui arrivent sans une valeur numérique dans le champ d'en-tête Content-Length, comme dans le suivant :

Content-Length: five

Les mises en œuvre devrait prendre des précautions supplémentaires si la technique qu'elles choisissent pour convertir ces champs ascii en une forme d'entier peut retourner une valeur négative. En particulier, le résultat ne doit pas être utilisé comme un compteur ou indice de matrice.

Détails du message : ncl

```
INVITE sip:user@exemple.com SIP/2.0
Max-Forwards: 254
To: sip:j.user@exemple.com
From: sip:caller@exemple.net;tag=32394234
Call-ID: ncl.0ha0isndaksdj2193423r542w35
CSeq: 0 INVITE
Via: SIP/2.0/UDP 192.0.2.53;branch=z9hG4bKkdjuw
Contact: <sip:caller@exemple53.exemple.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 49217 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
```

a=rtplib:31 LPC

3.1.2.4 Champs de demande scalaires avec valeurs surdimensionnées

Cette demande contient plusieurs valeurs de champ d'en-tête scalaires qui sont en dehors de leur gamme légale.

- o le numéro de séquence CSeq est > 2**32-1.
- o la valeur de Max-Forwards est > 255.
- o la valeur de Expires est > 2**32-1.
- o la valeur d'expiration du paramètre Contact est > 2**32-1.

Un élément qui reçoit cette demande devrait répondre avec un 400 Mauvaise demande du fait de l'erreur de CSeq. Si seulement le champ Max-Forwards avait été erroné, l'élément pourrait choisir de traiter la demande comme si le champ était absent. Si seulement les valeurs d'expiration étaient en erreur, l'élément pourrait les traiter comme si elles contenaient les valeurs par défaut pour l'expiration (3600 dans ce cas).

D'autres champs de demande scalaires qui peuvent contenir des valeurs aberrantes incluent, sans s'y limiter, la valeur de Contact q, la valeur de l'horodatage, et le paramètre Via ttl.

Détails du message : scalar02

```
REGISTER sip:exemple.com SIP/2.0
Via: SIP/2.0/TCP host129.exemple.com;branch=z9hG4bK342sdf0i3
To: <sip:user@exemple.com>
From: <sip:user@exemple.com>;tag=239232jh3
CSeq: 36893488147419103232 REGISTER
Call-ID: scalar02.23o0pd9vanlq3wnrlnewofjas9ui32
Max-Forwards: 300
Expires: 1<repeat count=100>0</repeat>
Contact: <sip:user@host129.exemple.com>
;expires=280297596632815
Content-Length: 0
```

3.1.2.5 Champs de réponse scalaires avec valeurs surdimensionnées

Cette réponse contient plusieurs valeurs de champ d'en-tête scalaires qui sont en dehors de leur gamme légale.

- o le numéro de séquence CSeq est > 2**32-1.
- o le champ Retry-After est d'une taille déraisonnable (noter que la RFC 3261 ne définit pas de gamme légale pour ce champ).
- o le champ Warning a une valeur d'avertissement de plus de 3 chiffres.

Un élément qui reçoit cette réponse va simplement l'éliminer.

Détails du message : scalarlg

```
SIP/2.0 503 Service Unavailable
<allOneLine>
Via: SIP/2.0/TCP host129.exemple.com
;branch=z9hG4bKzxxdiwo34sw
;received=192.0.2.129
</allOneLine>
To: <sip:user@exemple.com>
From: <sip:autre@exemple.net>;tag=2easdjfejw
CSeq: 9292394834772304023312 OPTIONS
Call-ID: scalarlg.noase0of0234hn2qofoaf0232aewf2394r
Retry-After: 949302838503028349304023988
Warning: 1812 ouverture "En cours"
Content-Length: 0
```

3.1.2.6 Chaîne entre guillemets non terminée dans un nom d'affichage

C'est une demande avec une citation non terminée dans le nom d'affichage du champ To. Un élément qui reçoit cette demande devrait retourner une erreur 400 Mauvaise demande. Un élément pourrait tenter de déduire un guillemet de terminaison et accepter le message. Cet élément doit s'assurer qu'il fait une déduction raisonnable quand cela se rencontre :

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

Détails du message : quotbal

```
INVITE sip:user@exemple.com SIP/2.0
To: "Mr. J. User <sip:j.user@exemple.com>
From: sip:caller@exemple.net;tag=93334
Max-Forwards: 10
Call-ID: quotbal.aksdj
Contact: <sip:caller@host59.exemple.net>
CSeq: 8 INVITE
Via: SIP/2.0/UDP 192.0.2.59;branch=z9hG4bKkdjuw39234
Content-Type: application/sdp
Content-Length: 152
```

```
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 49217 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.2.7 URI de demande enclos entre <>

Cette demande INVITE est invalide parce que l'URI de demande a été enclos dans des "<>".

Il est raisonnable de toujours rejeter une demande qui a cette erreur avec un code 400 Mauvaise demande. Les éléments qui tentent d'être libéraux avec ce qu'ils acceptent peuvent choisir d'ignorer les crochets. Si l'élément transmet la demande, il ne doit pas inclure les crochets dans les messages qu'il envoie.

Détails du message : ltgruri

```
INVITE <sip:user@exemple.com> SIP/2.0
To: sip:user@exemple.com
From: sip:caller@exemple.net;tag=39291
Max-Forwards: 23
Call-ID: ltgruri.1@192.0.2.5
CSeq: 1 INVITE
Via: SIP/2.0/UDP 192.0.2.5
Contact: <sip:caller@host5.exemple.net>
Content-Type: application/sdp
Content-Length: 159
```

```
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 49217 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.2.8 URI de demande SIP mal formé (LWS incorporés)

Cet INVITE a des LWS illégales dans l'URI de demande. Un élément qui reçoit cette demande devrait répondre par un code 400 Mauvaise demande. Un élément pourrait tenter d'ignorer la LWS pour les schémas (comme SIP) où le faire n'introduit pas d'ambiguïté.

Détails du message : lwsruri

```
INVITE sip:user@exemple.com;lr SIP/2.0
To: sip:user@exemple.com;tag=3xfe-9921883-z9f
From: sip:caller@exemple.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.exemple.net>
Content-Type: application/sdp
Content-Length: 159
```

```
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 49217 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.2.9 Plusieurs SP séparent les éléments de la ligne de demande

Cet INVITE a plusieurs caractères SP illégaux entre les éléments de la ligne de début. Il est acceptable de rejeter cette demande comme mal formée. Un élément libéral dans ce qu'il accepte peut ignorer ces caractères SP supplémentaires quand il traite la demande. Si l'élément transmet la demande, il ne doit pas inclure ces caractères SP supplémentaires dans les messages qu'il envoie.

Détails du message : lwsstart

```
INVITE sip:user@exemple.com SIP/2.0
Max-Forwards: 8
To: sip:user@exemple.com
From: sip:caller@exemple.net;tag=8814
Call-ID: lwsstart.dfknq234oi243099adsdfnawe3@exemple.com
CSeq: 1893884 INVITE
Via: SIP/2.0/UDP host1.exemple.com;branch=z9hG4bKkdjuw3923
Contact: <sip:caller@host1.exemple.net>
Content-Type: application/sdp
Content-Length: 150
```

```
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 49217 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.2.10 Caractères SP à la fin de la ligne de demande

Cette demande OPTIONS contient des caractères SP entre le champ SIP-Version et le CRLF terminant la ligne de

demande.

Il est acceptable de rejeter cette demande comme mal formée. Un élément libéral dans ce qu'il accepte peut ignorer ces caractères SP supplémentaires quand il traite la demande. Si l'élément transmet la demande, il ne doit pas inclure ces caractères SP supplémentaires dans les messages qu'il envoie.

Détails du message : trws

```
OPTIONS sip:remote-target@exemple.com SIP/2.0<hex>2020</hex>
Via: SIP/2.0/TCP host1.exemple.com;branch=z9hG4bK299342093
To: <sip:remote-target@exemple.com>
From: <sip:local-resource@exemple.com>;tag=329429089
Call-ID: trws.oicu34958239neffasdr2345r
Accept: application/sdp
CSeq: 238923 OPTIONS
Max-Forwards: 70
Content-Length: 0
```

3.1.2.11 En-têtes avec échappement dans l'URI de demande SIP

Cet INVITE est mal formé, car l'URI de demande SIP contient des en-têtes avec échappements.

Il est acceptable pour un élément de rejeter cette demande avec un code 400 Mauvaise demande. Un élément pourrait choisir d'être libéral dans ce qu'il accepte et ignorer les en-têtes à échappement. Si l'élément est un mandataire, les en-têtes échappés ne doivent pas apparaître dans l'URI de demande de la demande transmise (et très certainement ne doivent pas être traduits dans l'en-tête réel de la demande transmise).

Détails du message : escruri

```
INVITE sip:user@exemple.com?Route=%3Csip:exemple.com%3E SIP/2.0
To: sip:user@exemple.com
From: sip:caller@exemple.net;tag=341518
Max-Forwards: 7
Contact: <sip:caller@host39923.exemple.net>
Call-ID: escruri.23940-asdfhj-aje3br-234q098w-fawerh2q-h4n5
CSeq: 149209342 INVITE
Via: SIP/2.0/UDP host-of-the-hour.exemple.com;branch=z9hG4bKkdjuw
Content-Type: application/sdp
Content-Length: 150
```

```
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 49217 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.2.12 Zone horaire invalide dans le champ d'en-tête Date

Cet INVITE est invalide car il contient une zone horaire non GMT dans le champ d'en-tête SIP Date.

Il est acceptable de rejeter cette demande comme mal formée (mais un élément ne devrait faire cela que si le contenu du champ d'en-tête Date est réellement important pour son traitement). Un élément qui souhaite être libéral dans ce qu'il accepte pourrait ignorer cette valeur ainsi que si il ne va de toutes façons pas utiliser le champ d'en-tête Date. Autrement, il pourrait tenter d'interpréter cette date et l'ajuster au GMT.

La RFC 3261 définit explicitement la seule désignation de zone horaire acceptable comme "GMT". "UT", bien que

synonyme de GMT selon la [RFC2822], ne soit pas valide. "UTC" et "UCT" sont aussi invalides.

Détails du message : baddate

```
INVITE sip:user@exemple.com SIP/2.0
To: sip:user@exemple.com
From: sip:caller@exemple.net;tag=2234923
Max-Forwards: 70
Call-ID: baddate.239423mnsadf3j23lj42--sedfnm234
CSeq: 1392934 INVITE
Via: SIP/2.0/UDP host.exemple.com;branch=z9hG4bKkdjuw
Date: Fri, 01 Jan 2010 16:00:00 EST
Contact: <sip:caller@host5.exemple.net>
Content-Type: application/sdp
Content-Length: 150
```

```
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 49217 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtptime:31 LPC
```

3.1.2.13 Échec de l'inclusion de l'URI name-addr entre des <>

Cette demande REGISTER est mal formée. L'URI SIP contenu dans le champ d'en-tête Contact a un en-tête avec échappement, de sorte que le champ doit être en forme name-addr (ce qui implique que l'URI doit être enclos dans <>).

Il est raisonnable qu'un élément recevant cette demande réponde par un 400 Mauvaise demande. Un élément qui choisit d'être libéral dans ce qu'il accepte pourrait déduire les crochets angulaires car il n'y a pas d'ambiguïté dans cet exemple. En général, cela ne sera pas possible.

Détails du message : regbadct

```
REGISTER sip:exemple.com SIP/2.0
To: sip:user@exemple.com
From: sip:user@exemple.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@exemple.com?Route=%3Csip:sip.exemple.com%3E
l: 0
```

3.1.2.14 Espaces dans addr-spec

Cette demande est mal formée, car la spécification d'adresse dans le champ d'en-tête To contient des espaces. Les analyseurs qui reçoivent cette demande ne doivent pas échouer. Il est raisonnable de rejeter cette demande avec une réponse 400 Mauvaise demande. Les éléments qui tentent d'être libéraux peuvent ignorer les espaces.

Détails du message : badaspec

```
OPTIONS sip:user@exemple.org SIP/2.0
Via: SIP/2.0/UDP host4.exemple.com:5060;branch=z9hG4bKkdju43234
Max-Forwards: 70
From: "Bell, Alexander" <sip:a.g.bell@exemple.com>;tag=433423
To: "Watson, Thomas" < sip:t.watson@exemple.org >
```

Call-ID: badaspec.sdf0234n2nds0a099u23h3hnnw009cdkne3
Accept: application/sdp
CSeq: 3923239 OPTIONS
l: 0

3.1.2.15 Caractères non de jeton dans le nom d'affichage

Cette demande OPTIONS est mal formée, car les noms d'affichage dans les champs d'en-tête From et To contiennent des caractères de non jeton qui ne sont pas entre guillemets.

Il est raisonnable de toujours rejeter cette sorte d'erreur avec une réponse 400 Mauvaise demande.

Un élément peut tenter d'être libéral dans ce qu'il reçoit et déduire les guillemets manquants. Si cet élément est un mandataire, il ne doit pas propager l'erreur dans la demande qu'il transmet. Par conséquent, si les champs sont couverts par une signature, il ne sert à rien d'être libéral - le message devrait simplement être rejeté.

Détails du message : baddn

OPTIONS sip:t.watson@exemple.org SIP/2.0
Via: SIP/2.0/UDP c.exemple.com:5060;branch=z9hG4bKkdjuw
Max-Forwards: 70
From: Bell, Alexander <sip:a.g.bell@exemple.com>;tag=43
To: Watson, Thomas <sip:t.watson@exemple.org>
Call-ID: baddn.31415@c.exemple.com
Accept: application/sdp
CSeq: 3923239 OPTIONS
l: 0

3.1.2.16. Version de protocole inconnue

Pour un élément qui met en œuvre la [RFC3261], cette demande est mal formée à cause de son numéro de version majeure. L'élément devrait répondre à la demande avec une erreur 505 Version non prise en charge.

Détails du message : badvers

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

3.1.2.17 Discordance de méthode de ligne de début et de CSeq

Cette demande a des valeurs discordantes pour la méthode dans la ligne de début et le champ d'en-tête CSeq. Tout élément qui reçoit cette demande va répondre avec 400 Mauvaise demande.

Détails du message : mismatch01

OPTIONS sip:user@exemple.com SIP/2.0
To: sip:j.user@exemple.com
From: sip:caller@exemple.net;tag=34525
Max-Forwards: 6
Call-ID: mismatch01.dj0234sxdf13
CSeq: 8 INVITE
Via: SIP/2.0/UDP host.exemple.com;branch=z9hG4bKkdjuw
l: 0

3.1.2.18 Méthode inconnue avec discordance de méthode de CSeq

Ce message a une méthode inconnue dans la ligne de début, et une étiquette de méthode CSeq qui ne correspond pas.

Tout élément recevant cette réponse devrait répondre par un 501 Non mis en œuvre. Un 400 Mauvaise demande est aussi acceptable, mais le choix d'un 501 (particulièrement chez les mandataires) a de meilleures caractéristiques pour la suite.

Détails du message : mismatch02

```
NEWMETHOD sip:user@exemple.com SIP/2.0
To: sip:j.user@exemple.com
From: sip:caller@exemple.net;tag=34525
Max-Forwards: 6
Call-ID: mismatch02.dj0234sxdf13
CSeq: 8 INVITE
Contact: <sip:caller@host.exemple.net>
Via: SIP/2.0/UDP host.exemple.net;branch=z9hG4bKkdjuw
Content-Type: application/sdp
l: 138
```

```
v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.1
c=IN IP4 192.0.2.1
m=audio 49217 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.2.19. Code de réponse trop grand

Cette réponse a un code de réponse de plus de 699. Un élément recevant cette réponse devrait simplement l'éliminer.

Détails du message : bigcode

```
SIP/2.0 4294967301 il vaudrait mieux ne pas casser le receveur
Via: SIP/2.0/UDP 192.0.2.105;branch=z9hG4bK2398ndaoe
Call-ID: bigcode.asdof3uj203asdnf3429uasdhfas3ehjasdfas9i
CSeq: 353494 INVITE
From: <sip:user@exemple.com>;tag=39ansfi3
To: <sip:user@exemple.edu>;tag=902jndnke3
Content-Length: 0
Contact: <sip:user@host105.exemple.com>
```

3.2 Sémantique de la couche de transaction

Cette section contient des essais qui exercent l'analyseur et la logique de couche de transaction d'une mise en œuvre.

3.2.1 Identifiant de transaction manquant

Cette demande indique la prise en charge des identifiants de transaction du style de la RFC 3261 en fournissant le préfixe z9hG4bK au paramètre de branche, mais il ne fournit pas d'identifiant. Un analyseur ne doit pas lâcher quand il reçoit ce message. Un élément qui reçoit cette demande pourrait rejeter la demande avec une réponse 400 (de préférence sans état, car d'autres demandes provenant de la source vont probablement avoir aussi un paramètre de branche mal formé) ou il pourrait revenir à l'identifiant de transaction de style RFC 2543.

Détails du message : badbranch

```
OPTIONS sip:user@exemple.com SIP/2.0
```

```
To: sip:user@exemple.com
From: sip:caller@exemple.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 Sémantique de la couche d'application

Cette section contient des essais qui exercent l'analyseur et la logique de couche d'application d'une mise en œuvre.

3.3.1 Champs d'en-tête exogés manquants

Cette demande ne contient pas de champs d'en-tête Call-ID, From, ou To. un élément qui reçoit ce message ne doit pas échouer à cause des informations manquantes. Idéalement, il va répondre avec une erreur 400 Mauvaise demande.

Détails du message : insuf

```
INVITE sip:user@exemple.com SIP/2.0
CSeq: 193942 INVITE
Via: SIP/2.0/UDP 192.0.2.95;branch=z9hG4bKkdj.insuf
Content-Type: application/sdp
l: 152
```

```
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 49217 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtptime:31 LPC
```

3.3.2 URI de demande avec schéma inconnu

Cette demande OPTIONS contient un schéma d'URI inconnu dans l'URI de demande. Un analyseur doit accepter cela comme demande SIP bien formée. Un élément qui reçoit cette demande va la rejeter avec une réponse 416 Schéma d'URI non pris en charge. Certaines mises en œuvre anciennes tentent de chercher le contenu du champ d'en-tête To pour déterminer comment acheminer cette sorte de demande. C'est une erreur. En dépit du fait que le champ d'en-tête To et l'URI de demande se ressemblent fréquemment dans les messages de premier bond les plus simples, le champ d'en-tête To ne contient pas d'informations d'acheminement.

Détails du message : unkscm

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

3.3.3 URI de demande avec schéma connu mais atypique

Cette demande OPTIONS contient un URI de demande avec un schéma enregistré par l'IANA qui n'apparaît pas

couramment dans les URI de demande des demandes SIP. Un analyseur doit accepter cela comme demande SIP bien formée.

Si un élément n'accepte jamais ce schéma comme significatif dans un URI de demande, il est approprié de le traiter comme inconnu et de retourner une réponse 416 Schéma d'URI non pris en charge. Si l'élément peut accepter des URI avec ce schéma, alors un 404 Pas trouvé est approprié pour les URI qu'il n'accepte pas.

Détails du message : novelsc

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

3.3.4 Schémas d'URI inconnus dans les champs d'en-tête

Ce message contient des schémas enregistrés dans les champs d'en-tête To, From, et Contact d'une demande. Le message est syntaxiquement valide. Les analyseurs ne doivent pas échouer en recevant ce message.

Les mandataires devraient traiter ce message comme ils le feraient de toute autre demande pour cet URI. Un registraire va rejeter cette demande avec une réponse 400 Mauvaise demande, car le champ d'en-tête To: est obligé de contenir un URI SIP ou SIPS comme AoR.

Détails du message : unksm2

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

3.3.5 Proxy-Require et Require

Cette demande teste la mise en œuvre appropriée des mécanismes d'extension Proxy-Require et Require de SIP.

Tout élément qui reçoit cette demande va répondre avec un code 420 Mauvaise extension, contenant un champ d'en-tête non pris en charge qui fait la liste des caractéristiques provenant du champ d'en-tête Require ou Proxy-Require, selon le rôle dans lequel l'élément répond.

Détails du message : bext01

```
OPTIONS sip:user@exemple.com SIP/2.0
To: sip:j_user@exemple.com
From: sip:caller@exemple.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.exemple.com;branch=z9hG4bKkdjuw
Content-Length: 0
```

3.3.6 Type de contenu inconnu

Cette demande INVITE contient un corps de type inconnu. Il est syntaxiquement valide. Un analyseur ne doit pas échouer quand il le reçoit.

Un mandataire qui reçoit cette demande va la traiter comme toute autre INVITE. Un point d'extrémité qui reçoit cette demande va la rejeter avec une erreur 415 Type de support non pris en charge.

Détails du message : invut

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

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

3.3.7 Schéma d'autorisation inconnu

Cette demande REGISTER contient un champ d'en-tête Authorization avec un schéma inconnu. La demande est bien formée. Un analyseur ne doit pas échouer en la recevant.

Un mandataire va traiter cette demande comme tout autre REGISTER. Si il transmet la demande, il va inclure ce champ d'en-tête Authorization non modifié dans les messages transmis.

Un registraire qui ne se soucie pas de l'authentification par défi-réponse va simplement ignorer le champ d'en-tête Authorization, traitant cet enregistrement comme si le champ n'était pas présent. Un registraire qui tient compte de l'authentification par défi-réponse va rejeter cette demande avec un 401, produisant un nouveau défi avec un schéma qu'il comprend.

Les points d'extrémité qui choisissent de ne pas agir comme registraire vont simplement rejeter la demande. Un code 405 Méthode non admise est approprié.

Détails du message : regaut01

```
REGISTER sip:exemple.com SIP/2.0
To: sip:j.user@exemple.com
From: sip:j.user@exemple.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: PersonneNeConnaitCeSchéma opaque-data=ici
Content-Length:0
```

3.3.8 Valeurs multiples dans des champs exigés d'une seule valeur

Le message contient une demande avec plusieurs valeurs de Call-ID, To, From, Max-Forwards, et CSeq. Un élément recevant cette demande ne doit pas échouer.

Un élément recevant cette demande va répondre avec une erreur 400 Mauvaise demande.

Détails du message : multi01

```
INVITE sip:user@company.com SIP/2.0
Contact: <sip:caller@host25.exemple.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@exemple.com;tag=3413415
To: sip:user@exemple.com
To: sip:autre@exemple.net
From: sip:caller@exemple.net;tag=2923420123
Content-Type: application/sdp
l: 154
Contact: <sip:caller@host36.exemple.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 49217 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtptime:31 LPC
```

3.3.9 Valeurs multiples de Content-Length

Plusieurs valeurs en conflit de champ d'en-tête Content-Length apparaissent dans cette demande.

Du point de vue du tramage, cette situation est équivalente à une valeur invalide de Content-Length (ou pas de valeur du tout). Un élément recevant ce message devrait répondre par une erreur. Cette demande apparaît sur UDP, de sorte que le reste du datagramme peut être simplement éliminé. Si une demande comme cela arrive sur TCP, l'erreur de tramage n'est pas récupérable, et la connexion devrait être close.

Détails du message : mcl01

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

Il n'y a pas moyen de savoir combien d'octets sont supposés être ici.

3.3.10 Réponse 200 OK avec valeur de champ d'en-tête Via Broadcast

Ce message est une réponse avec une seconde valeur de champ d'en-tête Via envoyée contenant 255.255.255.255. Le message est bien formé ; les analyseurs ne doivent pas échouer en le recevant.

Selon la [RFC3261], un point d'extrémité recevant ce message devrait simplement l'éliminer.

Si un mandataire suit aveuglément les règles normales de traitement de réponse, il va transmettre cette réponse à l'adresse de diffusion. Pour se protéger contre cela comme voie d'attaque, les mandataires devraient éliminer de telles réponses.

Détails du message : 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@exemple.com;tag=11141343
To: sip:user@exemple.edu;tag=2229
Content-Length: 154
Content-Type: application/sdp
Contact: <sip:user@host28.exemple.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 49217 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.3.11 Max-Forwards de zéro

C'est une demande SIP légale avec la valeur de champ d'en-tête Max-Forwards réglée à zéro.

Un mandataire ne devrait pas transmettre la demande et devrait répondre 483 (Trop de bonds). Un point d'extrémité devrait traiter la demande comme si la valeur du champ Max-Forwards était encore positive.

Détails du message : zeromf

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

3.3.12 REGISTER avec un paramètre d'en-tête Contact

Cette demande register contient un contact où le paramètre "unknownparam" doit être interprété comme un contact-param et non un url-param.

Cette demande REGISTER devrait réussir. La réponse ne doit pas inclure de "unknownparam" comme url-parameter pour ce lien. De même, "unknownparam" ne doit pas apparaître comme url-parameter dans un lien durant les recherches suivantes.

Le comportement est le même, bien sûr, pour tous les noms de paramètres connus de contact-param.

Détails du message : cparam01

```
REGISTER sip:exemple.com SIP/2.0
```

Via: SIP/2.0/UDP saturn.exemple.com:5060;branch=z9hG4bKkdjuw
 Max-Forwards: 70
 From: sip:watson@exemple.com;tag=DkfVgjkrtMwaerKKpe
 To: sip:watson@exemple.com
 Call-ID: cparam01.70710@saturn.exemple.com
 CSeq: 2 REGISTER
 Contact: sip:+1972555222@gw1.exemple.net;unknownparam
 l: 0

3.3.13 REGISTER avec paramètre URL

Cette demande register contient un contact où l'URI a un paramètre inconnu. Le register devrait réussir, et une restitution suivante de l'enregistrement doit inclure "unknownparam" comme un url-parameter. Le comportement est le même, bien sûr, pour tous les noms connus de url-parameter.

Détails du message : cparam02

REGISTER sip:exemple.com SIP/2.0
 Via: SIP/2.0/UDP saturn.exemple.com:5060;branch=z9hG4bKkdjuw
 Max-Forwards: 70
 From: sip:watson@exemple.com;tag=838293
 To: sip:watson@exemple.com
 Call-ID: cparam02.70710@saturn.exemple.com
 CSeq: 3 REGISTER
 Contact: <sip:+1972555222@gw1.exemple.net;unknownparam>
 l: 0

3.3.14 REGISTER avec en-tête URL échappé

Cette demande register contient un contact où l'URI a un en-tête avec échappement.

Le register devrait réussir, et une restitution ultérieure de l'enregistrement doit inclure l'en-tête Route échappé dans l'URI de contact pour ce lien.

Détails du message : regescrt

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

3.3.15 Offre Accept inacceptable

Cette demande indique que la réponse doit contenir un corps d'un type inconnu. En particulier, comme le champ d'en-tête Accept ne contient pas application/sdp, la réponse ne peut pas contenir un corps. Le receveur de cette demande pourrait répondre avec un code 406 Non acceptable, avec un avertissement 399 indiquant qu'une réponse ne peut pas être formulée dans les formats offerts dans le champ d'en-tête Accept. Il est aussi approprié de répondre par un code 400 Mauvaise demande, car tous les agents d'utilisateur SIP qui prennent en charge INVITE sont obligés de prendre en charge application/sdp.

Détails du message : sdp01

INVITE sip:user@exemple.com SIP/2.0
 To: sip:j_user@exemple.com

```
Contact: <sip:caller@host15.exemple.net>
From: sip:caller@exemple.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: 150
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 49217 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtptime:31 LPC
```

3.4 Rétro compatibilité

3.4.1 INVITE avec la syntaxe de la RFC 2543

Ceci est un message légal selon la RFC 2543 (et plusieurs versions bis) qui devrait être accepté par les éléments de la RFC 3261 qui veulent conserver la rétro compatibilité.

- o Il n'y a pas du tout de paramètre de branche sur la valeur du champ d'en-tête Via.
- o Il n'y a pas d'étiquette From.
- o Il n'y a pas de Content-Length explicite. (Le corps est supposé être tous les octets dans le datagramme après la ligne nulle.)
- o Il n'y a pas de champ d'en-tête Max-Forwards.

Détails du message : inv2543

```
INVITE sip:UserB@exemple.com SIP/2.0
Via: SIP/2.0/UDP iftgw.exemple.com
From: <sip:+13035551111@ift.client.exemple.net;user=phone>
Record-Route: <sip:UserB@exemple.com;maddr=ss1.exemple.com>
To: sip:+16505552222@ss1.exemple.net;user=phone
Call-ID: inv2543.1717@ift.client.exemple.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 49217 RTP/AVP 0
```

4. Considérations sur la sécurité

Le présent document montre des exemples NON NORMATIFS d'établissement de session SIP. Les considérations sur la sécurité de la [RFC3261] s'appliquent.

Les analyseurs doivent considérer avec attention les conditions limites et les entrées malveillantes comme faisant partie de leur conception. Des attaques contre de nombreux systèmes de l'Internet utilisent des entrées élaborées pour causer des comportements indésirables de la part des mises en œuvre. Beaucoup des messages du présent document sont conçus pour provoquer une mise en œuvre d'analyseur sur les points utilisés traditionnellement pour de telles attaques. Cependant, le

présent document ne tente pas d'être exhaustif. Il devrait être considéré comme une graine pour stimuler les réflexions et les développements, et pas simplement comme un ensemble d'essais à effectuer.

5. Remerciements

La relecture détaillée finale de ce document a été effectuée par Diego Besprosvan, Vijay Gurbani, Shashi Kumar, Derek MacDonald, Gautham Narasimhan, Nils Ohlmeier, Bob Penfield, Reinaldo Penno, Marc Petit-Huguenin, Richard Sugarman, et Venkatesh Venkataramanan.

Les versions antérieures de ce document ont été relues par 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, et Tom Taylor.

Merci à Cullen Jennings et Eric Rescorla de leur contribution aux sections multipart/mime de ce document et de leur travail sur la construction des exemples de S/MIME dans la [RFC6216]. Merci à Neil Deason qui a contribué à plusieurs messages et à Kundan Singh qui a effectué la validation de l'analyseur des messages dans les versions antérieures.

Les personnes suivantes ont fourni des commentaires significatifs durant les premières phases du développement de ce document : Jean-Francois Mule, Hemant Agrawal, Henry Sinnreich, David Devanatham, Joe Pizzimenti, Matt Cannon, John Hearty, toute l'équipe de conception MCI IPOP, Scott Orton, Greg Osterhout, Pat Sollee, Doug Weisenberg, Danny Mistry, Steve McKinnon, et Denise Ingram, Denise Caballero, Tom Redman, Ilya Slain, Pat Sollee, John Truetken, et d'autres de MCI, 3Com, Cisco, Lucent, et Nortel.

6. Références pour information

- [RFC2822] P. Resnick, "[Format de message Internet](#)", avril 2001. (*Remplace la RFC0822, STD 11, Remplacée par RFC5322*)
- [RFC3261] J. Rosenberg et autres, "SIP : [Protocole d'initialisation de session](#)", juin 2002. (*Mise à jour par 3265, 3853, 4320, 4916, 5393, 6665, 8217, 8760*)
- [RFC3264] J. Rosenberg et H. Schulzrinne, "[Modèle d'offre/réponse](#) avec le protocole de description de session (corps)", juin 2002. (*P.S. ; MàJ par RFC8843*)
- [RFC3986] T. Berners-Lee, R. Fielding et L. Masinter, "[Identifiant de ressource uniforme](#) (URI) : Syntaxe générique", STD 66, janvier 2005. (*P.S. ; MàJ par RFC8820*)
- [RFC6216] C. Jennings, K. Ono, R. Sparks, B. Hibbard, "Exemple de flux d'appels utilisant les mécanismes de sécurité du protocole d'initialisation de session (SIP)", avril 2011. (*Information*)

Appendice A. Archive au bit près de chaque message d'essai

Le bloc de texte suivant est une archive TAR codée, compressée en gzip, de fichiers qui représentent chacun des exemples de message présentés dans la Section 3.

Pour récupérer intacte l'archive compressée, le texte de ce document peut être passé en entrée au script Perl suivant (le résultat devrait être redirigé sur un fichier ou transféré à "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

Autrement, le bloc codé en base-64 peut être édité manuellement pour supprimer les lignes de structure du document et l'entrer dans un utilitaire de décodage base-64.

A.1 Messages de référence codés

-- DÉBUT D'ARCHIVE DE MESSAGE --

```

H4sIAEDwcEMCA+xdW2zc2Hm2nexNG6UN3LRF0QfaiKJdyxwdnkMOhyOPVrIt27It22tdvHYTeM8MDzWc4ZAjkqO
RvK2bbIAAedmHtEHRdlvkoUCLFAjSlyLF9rJPLYoWrTdAg6JFHwp0i+5D0SIoEAQFuj2HnAuH5GgoW3PxmngcazYU/
b4f///3Xc04Rq9ipk1JGxe6xITVAW1YUvXc5K/W8syYheEygP0IIECWJ0gkSkMAxDhwBQW4LrY57phdcerYrjr96AZtb
91L5/0paTdvbavezLHOOXo933CivUT2cK1ukIxl3Prq7fmYQZMT23pOn/
+Nr958RZXthxXzLRpS1YtL4EsWCja2CyVCw+U8mWxeK2qVhoigkicnlrDe/wly25iW3XynEwPecmmO3GnxxPDOMst
G/RQpkr09w5diU4s50p0i1LgTMsLrh4uyAiJEI0PbVh0Z3vYNexzLpCRtmqYYfu692Gm2l6v/fcyuL01AVsGPzqxTxXbP
O8o2qAXp4JTdUBGChKA6IyKptmEwClpFZNQs+0XCqRupvncL1u6CXs6pY576h1erx1spPnkALpLSpcqyOnp4w8R29
veurpeP60L/yHnkQAGCT/IsiG5R8KMJX/sco/lbiu/DNpi6NoizHbVqLi1YsfnsAiBEUYBgAUymCQj9IYEYIochBEhiQ
6BYXO1i1TM2CSBchqOwC7AAKKxqqILMtsbmnVvaHJP9U8Mkw1f8g+RcAFI+xf6IIBJRFV7FbDbV/yNpqqe2Vjd
Wjl78TeJ64g+pfWYwo5aAEHp9XiQqlGq22smlWEqsBAZFRHyvENUzax5VoQvwwJVuQoSof/S+XgnQdskxixpTk9dp
KfMc5ds/SwHBO4qNjllWZETsnkA6B+3sr5Bz2iZLi5R7DkXuEd2fCkTuNNFn5CYLr+XydxSNXafJ2Y226Z3oPk4c5u
gb5ZhVqZGj8G2eegllNTQoYyvUGF3iC3ekvsAKM0PeUU+OmpUiG6wS0AhmS1Am6ousXRLhdk7vbGrfnlrVscvSnltu
3qKrE4BGF3ExKmp3DBdus1XM8jgbt+68KzjlbPjv6bQ0X/BP6fgEL2n4gkKcX/Udt/sY5Trw/IWhBqS0l8wGYY/b3WdQ
JpC7mBg71AXyl5rdcl9HeNu5WQC0jZnvKbIC313MNAf4+2Pi7f0yr/urkLhHHGf2QEwvafDFAq/xNn/1Uyj2EBckgUst
SiF6CYjZiBvSLpicyoY6A7poqrjlrBDrUFWYwGO13/ra/11/EhpYWFswtnzwYMuNNXLdKkLIUs0oMLC7TFmWhwoF1
3SAto6GvCCeOy4Wiv7xLbGaf/B0QxqP8IT/5RNpX/idH/ckT/y3H6P6nq79H8yxlp+Q/S+DtNYuk7dRLQ+XuzlupPrPI9T
mdKXw4r/Y5uF56x4FCxhKzmzPFb7n4p+JP6Dsqn8j6S1tCeHAeBuXjtIpSq5kHwLCPqhncg+UJiygVd4PwcXic127Mqmx
4UA5cScCCAQqMKnyl/DVVSbXG4SVXOW11Wtk3OROiZA1/wImya+8SFQaUdtdyFCrTRGK0oFITgLEwU2OkGiL
yDs/AQzAXzZfHwloKS62sqse1pvCdtR4P/ZM8drveXIP4jS2H7T4RCiv+TI/
+r3H+cFIAliWuHdcFsEP59Juxx/KanbpOdhm5T0DUtt6yb2+uNet2yXWejrDtn435c0d0yoSe6ZVt7+3xgd/aDTpwWbXt/
+6K1b05Ht8XkCXs03Mb1dU6zDJWnQM5T7vEUySAROKxbJsWyLORBjUSda/4PSIIY8f/S+M9o7T8RKqKSIREQqDS6
LrGZgHFFm+AqR6Wks0mJ6LtmuvpbiCBs6UGk5K4WYqo6y2Gw45qaahRgQDRj6aG6BU06Keyhh1Eyl7gBzuK3rX5
kKiilbvXbxs+Q4jUrDpaHrL8jsXZ/r5hAqAFVM1q6zWJ0ZK+Xh49GYjFt7TqP9LFLHtMed/5CwM+3/UaE/lf2Liv72aO/
ekEWGF5fnpPww2S7A3zG17P5xhbyOlz719xkKT6JiHvPFCYLEven7qMbmWX5H5CGSIDp0Y1+h7THiwgcEhocbGS5
Rpsa18eZ/JCBE9b+cyv8o2u2Vy6vrGyu3PXmNFf6l/DjYbdjmyYv+u5FfdrpQvLYds7IY1ba2K1XbXWtiY1+71g76xu8SB
IY2L1PuEcBS9Drb4AC59m0HAIGdfk5QZEhZENK2tN0UghC00DCrptU0vZN8YqLDrT6D45R/MStH5B+m+v9Zlf8cyl
EleTiZhwNIHsXJ/LIDCF3iJzAnpBYNm+ymNf4nICkbtv+ltP5rUuQ/makf3doq1IKSUEEVBCODgHLTU6t5rsV/Pda8ojDf
XzMNZFQhQqIAQ2q6dbJwnW/X9u+Kev9oBZTiEMsrV+4TrqMX3PWWgkUkPf3Vvsbe7UkKZajTi+K6qQN24c5S5ZJn
KleJJ01KwloQwhTlxnYBywK+3Hn5R85OXxHCJPZ800xVtxCkNO/0zOP+P5DD+wzT+M/L4D305M2iZQeuMCALYF
USqqF6YkSWHzMgwDu08+2p1BLE2iX0jXZhiTjB47VisCgXwT15ekrPcz5/spEhQPFcwtVK1YTIMukXwKPA5BDIBo
aKScM+DHTjEzUHJPmnp7hOnGomeyFuKMgC/Z12xoI5kxVqpLbL34wGvXVpBokzSFg8ItTsC5ppaUDaDov/cMzx/z
U/6eSnOL/aOz/GVGmtvKMKPk+gE22dce1sZ3p6w2cnrlHyVvF1DZxrIZd6jF2Fzvd+wdSevCvQQQRWNUqPUVhPmsy
0Dd3mhYS7R2IdCRWbJYbLJkGRKZtVE2H1YX1JugvDBwDCIEyoz1wTGIENYiCRJEL9kjssMWe4AE2dOwIfkowlB
C8MJO9FCGFfnyMdr6TOQRohDhkccfaCebDcMYb/5fjMb/sun4/wnz/xmb8DMA8OxDP9e2L1ErsqcoJ+/44DhwG2
WRAoEYFS13WpFxy5W3UFN0XLtHYBVR6OggHfVzpbGESk5Zv0d4PgPSvsFCnW6okhvZSmy42IMVT/C6/nJjhfSrz
Ybtj7e8f9iP4nS6n+H7X/Fw7gvXbbarikMIMuHLBhBq0cydwASBQkIRc3JqzfqHsPP/rVBbRZ2XMWEWY31Cs8rhBUtH
mKDTtAyTV5DTeJXY7fFk0pS58UKihyh8e7D3ylsa7ZcKXqRmTvOjvmMGz1A1M25M13XQa2pj9PzEbGf8rpnvseN
/F+NbKONvBq3OKSgxOYWmX2cDQdFoDbs9JA4qfZMISjo3yiD5d2vELY/V/oM99V9Z3/5L/b+RtFOUAYhNHf3ct/k1
ygmW6g2/k7JyTrl/Zu7NzIxuqoSosw89kBDuN8yG08BGZvP26sNXXIsvklN0wyavflF3zF13Tne/MF+y8YP9187OLyyey
X9Rd+fK2CkIZ5tEt9VTZ+rY+ULTeqjedy0r84pqEbYbr7uvLg0uP2IHdoSDyjcZp2ay2TP3596cfdiKV51fuZ7/0gvejU36do
y7yL79aisoddj7iQ1zuVaVmMLDN/0PcHbuvv8JnZk5leH9E9UaporNUPBL07vfYqULs7MP587cp8QzhdMffOXR9x790a
M//
+Dtr9/74J0PvvHo+4/+5LRnMN/3jvpQmJ17kx6ZzwSM37YcNy1bnbl3jT+VoT0wu8S+vvnw1VcWz+W/NH/6y7/02q+8
FZhGS4AQ5STuEDwQntYhK08lexTTHQriVwhWiU3PPXNm7j7t/vx/vfcXH/7e73/41Xc/+u33JncMzLnT/
+1CSURjjf9lvff/IfsvHf83avtk9p/55eS1QDqmrzdPtl+M4JCCBJkgTalihppmToVPB7C+vo2Qr1smWSRTbUr0SBivcC
DqljRK5moYZV1S44Tjj03o5AzAlZCbTr+IJKvafrAU2f+QVZkOOuM1BTJGU7hu/Rzgnz+LP6HQ120i5ouOPO/wE5bP+
J6fyfk+T/JZ0F84mGBeW8gL94YG7AZ9feGaJUR9MrBZvpHZrQSRRPKD8zbFqJNks0f2FvVUZrFl2DbtR40b0rJtznUT

```

SnuHO1Uu1BsfGGBdOiyI6PU9/PDff3jy2529Y5vawC4AHyH82K6NI/D/N/02Q/HtO1CrHirg4zDE6zsQ1wlkaR21/m9TI
E75xddtiokHl6nTsmN58lvZpTzG+UNgl9j5j36N87WKjQRbYJ+8k7C6H/So4ZXrn/omHcUtxL8/nJNTpu0Hf7uhu+Ya1xS
6AebQ+RuMafkDdQcO7HB+w2cUO8+doQTRUcpP4J0KxzYplz+MdCq8U/FMEzuDxyNH8a1+/99QN4jidWzjyDwHt3
VY21Aq3Cflxf/T/2yynd2wF1GOVA6BjLGz6PcNfp5RH+eKZrOW5VsfzRy3SvP9ch3f8ehszeA/7C6M4k3dPMTFAiA19
bpuu1EK2EuxFaUQQhRx5wFhmuUIDVRCNKvR4/TOCMZoYo4jh+4p5npgNkwTexwpRBN3PWKYJuY4oT7e4vJchCb
Uy7txNOF+5rjouUD4OFEaQYk8rxiiXJQohkoe/OiFbAIAkQGNmIAGJaCBCWgSsLQABtMog0lyg0kSdHKCPk7QxQ16
OEEHJ+nfASQRr7I1cY5a6MR12o7mqly9Ubz8W2rB9cCa2THYlo+xZofxLBTlGO62O+wCwIHZ/0Tmf5WetP5vpP5//E
RaR1Plp0BFiOQNq4X+HR4UlgLB71aWcnC9iTTMZxqUIw7o10FUEMzJYEKAwGg6qu7Ux5r/QzKL/7OB3jKbDNqP/
6X1XyNpR7P+ny9x52JQoDsf34Cq/zYjssIDXCypSxr1L/fjUnFZOGdiTgYKkb3iw/rpwn9d+R9//T8K5/8IANL5P8Yd/1/gD
HswBPjCvacRXqFKLJdD/ANFSzIrMPJnZo/k+6ReUPC4058MVLWlPbO1l7zi/qZt+p9ySLSr3qCiVvJPQNSkzIooQa2q0
HfqIoiUgwwLYfSGxcGOxaQZfml7WvCfAaA7Vv9PhhH/T07rP0aJ//H2X98ZYJ/IiczBLEX41scqFXNHWr9UYwXEV
AUrLKh37hJ0FKMFSjkFJTLDZjvQxhkCSKIPcfrcFB+4sNhTzIx7vl/gCRnw/Vf6fwPE+X/HXoyHTaVakTJYMJiqzbEh
Y3YcKMAUPisqpVNZgJatl7GTU21MLJEW4IW0m2nu0IQta+o+ddxEyOCGffFsyBKJYUF3iV77nzdWlrJxHqDXjaZdT
jT4pp4n3MtjuVDadc0uRo29zmr5BLX4bBNOIetLuEQIREVCcd2zEyG+1nTnRp2S+VhgsDA+18khuRfgmn95yTGfx6rrh
OJEpQOWv4lyIMVnVogs6dqRtKp3NNgz5HIPxyf/Mve+L8e+59+SeV/FO3Gyp211Y0rNy9OBgLAAQjQ11IPGeoHI0X/
yf8GZ4TZTIWHN+njrPmJiNHUKFMPewG4AflPRYH5/wjK9Mkj4M3/JKX6fzRtbWV9ffmyHwCuNmp61Yqs/hvVaqh
QpowMvAiqHI6g8mpOlkTEiypWCciaoKgWBYEQXuDbmxZsVvAdlxIKjgILniUHqGCvqlQ+dXc/z9lSrWmT6439i7fvnt
fnwYZ+q7ni3EX2ypZTnb8M7+yVtze2hFuX5PKVxvXVa+Duna1r98rXdpW1Sm6TvLFTXdsqms7SL5weS4noO3du5XS
+U18z9x+vfZGSblz4976xRuXLzxYntPuGMrayuaVO6v3LOECuUqK6t35N4zta2QDg5vG7g6wXRm8LIX0da24s3a19vol
lxB9d7NcXbtzd3O5EA YuzWho2v5Sbx8uBpaqij6KxZ4Rdq0DRLJsUCsCtVRSAtiKVEXMSVgFJVHJSUpX6V5t7hXuy
ddvyOU7+6DvRu65BThjXJxOZPppt1bjBHARJstqEVsfsUsWapubrO1vExs74dhs9YwXJ3BynxN3yPqQtFiQ5/tYKMIV
KRIAsEaJl2cjh5g96XXsnqUeG3ez5LJ1aCUOihIRzEtZ7vLaxvQUGwnIL2/TE3gcqNfqrD/mvZQ690qpYTvEeTXGe5
MQQ2qej562v1eX4Ganp64Qw7CSHDmoQDxvkHdem7DVQxKdB7x9/PbzL535+pWv/3j6+Asnvv328Qv0p9dOHD8uvA
ReeP65uc9+6sQvgJe7JMeFt49z1OQXKQn4ZVAXXgYvPf+pzU8/f/LE5rpwEnyOfXnxJOMNXbNsU8fC58BPsR9fOPniO
ja5q9RJFT4LPsN+mjr5HOU03RvEbbPs+8snuXX2ndsgjstlLara+zmCLfccMvUH3f3T7x4/FvHdk6g4ye7lzf2dubzjx5879u
mvfuWd/9F+cuW/b/M/
+PA33vmdb33nox/93+e/rd3dKX7zP/7pzB/8ozRV/ruPv3vuz+Y+/tnGv/yD+s7dwg8vv/KjIvfjj5bd30XTv/X5//zDf/61Dz5z/
qczb9nf+MG77//1dzP/+1fv//1Pfm5+4cTDM+/P/utG7d/+
+Dv59xa1v/2bL//wL6997Ws/8+73//3jn+d//eaHb5E//c1Hz8U9OJ5/Vr0DT1jHq/8FCILjv0Rf/6frP40l/0s1SR2b+wnHf0Jpk
N3dtmfV1ulrQjxx1t3Tf6Ok9BiVsxHloloClijSrV2UxPvAA5yWtk6bpaZigT9Y5jtLZEIZpL8tzcTJptn1lkhqjQtHjZ0GO
IZNQP6bHAvkVs8KgnVERDMmNtZskYcvYnQfynZ/5PD/9BWv83HvxPsP4nZFj0ZPEgL7WDgllgxc9w/qh4KWQbEm
ETdQpBwYJyoFRkol9e69MQk+W6OUfZ9VP5p/0qRRGIxsqYFrUIXEsc6zr/0EYyP/IXv0HSuV/JK29/jft+6J1/buMBOm
UxW9tbZ7PyKAiGbx7u7kne6ivfk6H9rlxhOaaz1/yKUQvKf1v+PPP9r4XqmSEg9Pz/fXqURwDAoxkGlltU+qle6PD50
K/0pYcHOxRhzKZf+F5l1uqt0IH8JgT5jANpjmJUKlV9iOifhTsNyi3i863/IohS2/+V0/c8Jsv+9+X/W7Ax3NcOxaLyyRaOm/
2ICHFKFcgEKXAnkUJslPTfgoLF9SthoT7rqaPtkJSCBBLCQuNATjmhly6PIEdtkGzeGnABONv97r/4X0/k/R9KGvP5v
TkZQKfCgEmDuoJvtO2zYewNoizTFJL5r+jKF3w1O9vH9WxmB54I57kb1k2TXD0tprNR1p31UplNQWbV8U6D8FQ
OclEVaEak/BNpGtCOL2K15I5X/IHY/wdp/mfS5P8g6Y+VfUXJIQQPHPTb4b4qEiXs2AbS1OLukgD8sohuNcJBC3ojKS
PkKDVil9R3QF9oDfGDVzuir0zvikef0DJS+gTYuknjX8pG43/pfWft7v85waJvs94IdEPZ3X/v72j2W3bBt8L9B2IALsEc
UqRki05c9Fi2LB2a7I169JdgUKxKfV+oRxSspqctp4H7F2G9bjtFZ13Gj9RdiXZlpN2tZNVbFLHICjJ9Pf/
+x+29nu2wtS4DvXvQBu/Dx5SKVSu+LQqtNfq/9RckP9Jf+3BP+XN2AhVQY2ahLp+eFqs33eVIBh/SegiZNBfx4ITpu2
Q01IIFUKogMxw5TkCMocuMyhmtAQTxzn6vLRGU24GHGWhP7AIU4cLMomFKvP+/WbSSBYapTYwFjtmijuMJQQ
RwdMP1uH2Jg4LctpNimxDesjzJh6p0a9beJ/U+d/ENLEoCim8n9t/9us/8/CFB0zMQ26DL3g7tQNRu7piH0Q+I9cvPEC6N
gk8Xyh2rptanCQlxVRhaYK72BzwbJ3EBA0mxRm5qtfqHYhEQCI5//WSQMctx56EqGQx+STfqnIV+6MlB4rLEh2sJ6
EoSifPGYz9iQt3CdCgmNrXVTsErvIX70LaSc05cWdPgXcNwTwt2nTESxYgjnCUc/iLCnUFbu3E5fg/QmW8//blkL8T+E
1vr/7bL/p0a+14tKwOraCeXY0GuUBKosBqRBNTMNQnQCP8iAd//e426XTaIsjp+Hp6F3Prft3cApYFjVOWGFog63pQf
QRzsCJBsHW67/ha1WTv5vZvX/mjX+b2KU878P1IfsxF+YOIecNycKK1E9FD0dDUCJSariAWZwudQfYL/n7DPkL6Hg
XmFiMuzKVmMcjNkeiNF6lNKU9nITMuhxNZM7VzFwt8fUAXX13BtfuL0qknL789EjJ91+6/19r98/ReN/6RV+/82zf+V
ZMwfKdU4HADzESMRUohVntW4LdzRPF9E+mfuy4Z/Je7c596xgnDczDv4UjgrIHpe1Pfy/OLivEHUaYpnl/WjBH/E4/E
pE2olRzpCQJ9F23Nenk4UFHX1FMzVorgfhuhUfbXqa4OnDkN4I9KXJUzdvi5Pj3Sf1E6auHH37X8RJCBJq7cYv2HZq
uE/5ZV2/+3w/8/rP7vR1X+RTkYHAADvukOpe6eDYYJ4550pQ+MfwWfv1ZN4PVI4EqLjr/6Sa0iVatGrhx4fb9sH/n+GE
W0Yh1xbBxzDz5R8TEPvzs8OjIEkVmxWkK72KeMLzHKVLo/sOsEns8ZWyJ6RCKR2+b/Fi7jv0lxj8fbx3/BxmHEGmo/
esuFfoQq+7GvgHhQ1bGT9wWsvFHR/DcKFUlpCCbDWHtZMsLimMTBdr7aQQrPYdCNqelYNoQQMx8iiPsCGpLTY
SEUNvJBxcv72Xy/4kYjvQdseflgKslf9NWsb/Zo3/m8X/kt1Mx8S1ozCCpnk6NK6rAX8T2QAZVPI6G2AT+D8mW8Z/S
sr4b5Fa/78l/n/A8kCe8raSXWnLIXaLzrl0P4ogYShJkv11Yr9CzDJWp7CWqv/980mfcTdiXqNcxvE9MzYdnHfwL3Lj1QU
njOteAn4JzeHaT8M+f308DqL+w8+tWmTm57/a2Cj7P+zDfRxf9vImEn2BGPUQWR3l6JdZ03uEnT51+Xf6OrXq7fq5f
LPy3eXf1z9DnOlqau3V7+hBrr85+qX2bHLd1U2cccu4aRBIemQIqHIGBKAQ2B6Q/pgDCRMM+Xj48rUolXBIMahm43
skJqYV6sXZAEwhMWXX7XSwlaFsbj2AsZxdLE3LslXICoJ2tRSsx0aiQz4dLvxH0bLMSrxH0oArOn/hv0/02DKhr1H
3b4biIbHRbegNGXdvtPe3Dnh8Ai1799bvxdIIQUgQVnakeRMMOGf5Ty+/d8AZ1GUBvtPEXPQ8n4KRM99OrVq52dz
Puqs009AQcflcREwn9w5Q6CII8Dz17jidIQv8QJVaT66SOLR5dmVaewdiBW+XiSLK20gg9UCen24GA1Wm38VwOnX
E76mCZiZ8Snu2QJd+4vDdi3rfM9SEDEDCHOEl/POayXLgKtO+Cxmllk8mLWq1+tpIPj6gscydke/w+E2OjjY21O/B2t5l66
4qm6W7rTUYCY8PWPxAwCEnWfjm42P+sAz0Pdqlh2oZ9tiepwFABAh0chpwEwO4KUJF9fXqHADIQyoQCJxgObI

TABz/fYn1F9SfIE+kGG017n1jWvuLucK3sQh21aBaJDBF6igO2d36MQ6VqIBmJgCvw2gHwWglY6lJqtzWsd2ZhAGom
/ZT6mSXYQzx9ygE6U8+O9ym9MXtfWQPI3Axxv2AK/fwt6/8EL9r/av3/Dvn/qix9vb70TCNHgDOQG4Apb+TzMqnJyK
TCirE29xVMe5QYFZ69a/V4AitCULYd4Lr0Rz3qUY/PfPwLcaRGXQDwAAA=
-- FIN DE L'ARCHIVE DE MESSAGE --

Adresse des auteurs

Alan Hawrylyshen
Ditech Networks
200 - 1167 Kensington Cr. NW
Calgary, Alberta T2N 1X7
Canada
téléphone : +1.403.806.3366
mél : ahawrylyshen@ditech.com

Robert J. Sparks (editor)
Estacado Systems
mél : RjS@estacado.net

Alan Johnston
Avaya
St. Louis, MO 63124
mél : alan@sipstation.com

Jonathan Rosenberg
Cisco Systems
600 Lanidex Plaza
Parsippany, NJ 07052
téléphone : +1 973 952 5000
mél : jdrosen@cisco.com
URI : <http://www.jdrosen.net>

Henning Schulzrinne
Columbia University
Department of Computer Science
450 Computer Science Building
New York, NY 10027
téléphone : +1 212 939 7042
mél : hgs@cs.columbia.edu

Déclaration complète de droits de reproduction

Copyright (C) The Internet Society (2006).

Le présent document est soumis aux droits, licences et restrictions contenus dans le BCP 78, et à www.rfc-editor.org, et sauf pour ce qui est mentionné ci-après, les auteurs conservent tous leurs droits.

Le présent document et les informations contenues sont fournis sur une base "EN L'ÉTAT" et le contributeur, l'organisation qu'il ou elle représente ou qui le/la finance (s'il en est), la INTERNET SOCIETY et la INTERNET ENGINEERING TASK FORCE déclinent toutes garanties, exprimées ou implicites, y compris mais non limitées à toute garantie que l'utilisation des informations encloses ne viole aucun droit ou aucune garantie implicite de commercialisation ou d'aptitude à un objet particulier.

Propriété intellectuelle

L'IETF ne prend pas position sur la validité et la portée de tout droit de propriété intellectuelle ou autres droits qui pourraient être revendiqués au titre de la mise en œuvre ou l'utilisation de la technologie décrite dans le présent document ou sur la mesure dans laquelle toute licence sur de tels droits pourrait être ou n'être pas disponible ; pas plus qu'elle ne prétend avoir accompli aucun effort pour identifier de tels droits. Les informations sur les procédures de l'ISOC au sujet des droits dans les documents de l'ISOC figurent dans les BCP 78 et BCP 79.

Des copies des dépôts d'IPR faites au secrétariat de l'IETF et toutes assurances de disponibilité de licences, ou le résultat de tentatives faites pour obtenir une licence ou permission générale d'utilisation de tels droits de propriété par ceux qui mettent en œuvre ou utilisent la présente spécification peuvent être obtenues sur répertoire en ligne des IPR de l'IETF à <http://www.ietf.org/ipr>.

L'IETF invite toute partie intéressée à porter son attention sur tous copyrights, licences ou applications de licence, ou autres droits de propriété qui pourraient couvrir les technologies qui peuvent être nécessaires pour mettre en œuvre la présente norme. Prière d'adresser les informations à l'IETF à ietf-ipr@ietf.org.

Remerciement

Le financement de la fonction d'édition des RFC est actuellement fourni par l'activité de soutien administratif (IASA) de l'IETF.