

Comprendre les échanges SIP par l'expérimentation

Emin Gabrielyan
Christian Lathion (traduction)
2007-04-10
Switzernet Sàrl

Nous allons analyser quelques scénarios simples d'échange de messages SIP, pour un appel entre deux téléphones SIP. Nous utilisons SIP Express Router (SER) comme proxy, ainsi que pour analyser les messages SIP. Nous illustrons et analysons les requêtes INVITE, CANCEL, ACK, BYE ainsi que leur réponses. Nous discuterons les transactions et dialogues SIP. Nous examinerons le cas où le serveur proxy intermédiaire ne participe qu'à l'établissement de l'appel, ainsi que le cas où il modifie l'adresse des messages de signalisation pour participer à la signalisation de toute la session SIP. Nous n'examinerons pas le chemin du média. Lors de nos expérimentations, les paquets RTP sont directement transmis entre les deux téléphones SIP, sans intermédiaire.

Comprendre les échanges SIP par l'expérimentation	1
1. Configuration de test	1
2. Le proxy SIP ne prend en charge que l'établissement de l'appel. Les appareils SIP transmettent le reste de la signalisation directement.....	2
2.1. Vue générale du scénario	2
2.2. A quoi ressemblent les messages SIP ?.....	3
2.3. Configuration des appareils SIP.....	4
2.4. Configuration du serveur proxy SIP	5
2.5. Messages d'établissement d'appel.....	6
2.6. Appel annulé	9
2.7. Configuration de SER sans mémoire (stateless)	13
3. Le proxy SIP traite l'établissement de l'appel et tous les messages de signalisation suivants	14
4. Autres expérimentations pour comprendre le protocole SIP	21
5. Glossaire.....	22
6. Liens en rapport.....	22

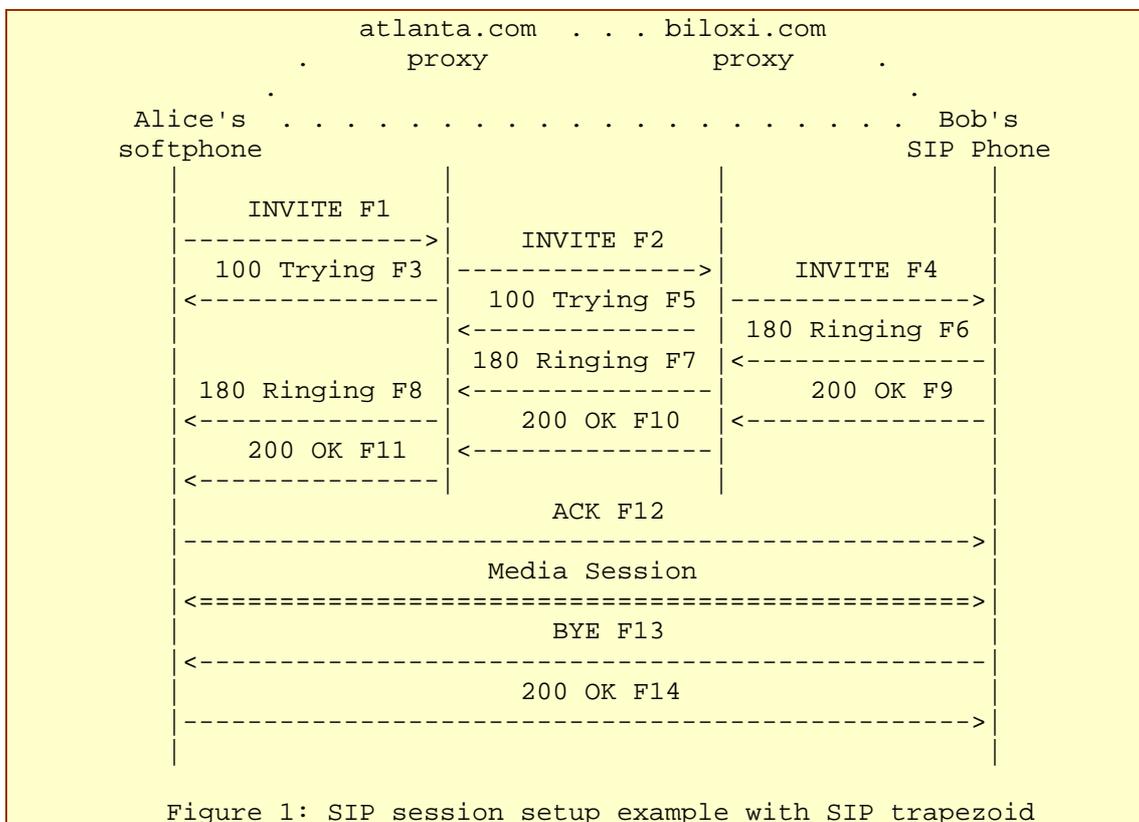
1. Configuration de test

Dans nos expérimentations, nous utiliserons deux téléphones SIP (Grandstream Budge Tone-100) [308, [bmp](#), [htm](#)], [309, [bmp](#), [htm](#)], ainsi qu'un serveur proxy SIP OpenSER, une déclinaison de SIP Express Router (SER).

2. Le proxy SIP ne prend en charge que l'établissement de l'appel. Les appareils SIP transmettent le reste de la signalisation directement

2.1. Vue générale du scénario

Ce scénario (le proxy SIP ne prend en charge la signalisation que lors de l'établissement de l'appel, le reste de la signalisation passe directement entre les deux téléphones SIP en contournant le proxy) est le premier exemple décrit dans le RFC3261 [txt], [htm]. Voir le schéma ci-dessous [rfc3261, p.10-11].



Dans l'exemple du RFC3261, il y a deux proxys intermédiaires. Les messages de signalisation SIP pendant l'établissement de l'appel sont échangés entre les téléphones SIP via les deux proxys intermédiaires. Les messages d'établissement d'appel sont INVITE, 100 (Trying), 180 (Ringing), et 200 (OK). L'échange de ces quatre messages constitue une **transaction SIP**. Il s'agit d'un court échange de messages SIP. Dans le contexte d'une transaction, les messages SIP utilisent le même chemin, donc si le message INVITE initial de cet exemple passe par deux proxys intermédiaires, tous les autres messages de la transaction vont aussi passer par les deux mêmes proxys [rfc3261, p.13-14]. Dans une même transaction, le chemin est conservé en utilisant une pile de champs "Via" ajoutés à l'en-tête des messages SIP [rfc3261, p.12].

De plus, dans le même exemple, on peut voir que d'une fois l'établissement de l'appel (initiée par un message INVITE) accomplie, les messages de signalisation suivants sont échangés directement entre les téléphones SIP. Le message ACK (informant du début de la session média) est transmis directement de l'appareil appelant à l'appareil appelé. La transaction de déconnexion, comprenant la requête BYE, est aussi acheminée directement sans passer par les deux serveurs proxy.

Dans ce document, nous analysons les transactions SIP en utilisant un modèle plus simple, comprenant un seul proxy intermédiaire.

Le chemin des paquets média n'est pas décrit dans ce document. Dans tous les exemples discutés, le flux média est transmis directement entre les deux téléphones, que la signalisation passe au travers d'un serveur proxy ou non.

2.2. A quoi ressemblent les messages SIP ?

Les messages SIP sont des lignes de texte. On peut les imprimer. Un message SIP est habituellement transporté dans un paquet UDP. On peut répéter une courte expérience transmettant des messages texte dans des paquets UDP [\[htm\]](#). Le protocole SIP est basé sur un modèle de transactions requête/réponse, ressemblant à du HTTP. Chaque transaction consiste en une requête et au moins une réponse. L'exemple du message INVITE, correspondant à la première figure du RFC3261 est illustré ci-dessous [\[rfc3261, p.10-11\]](#).

```
INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bK776asdhds
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 314159 INVITE
Contact: <sip:alice@pc33.atlanta.com>
Content-Type: application/sdp
Content-Length: 142
```

(Alice's SDP not shown)

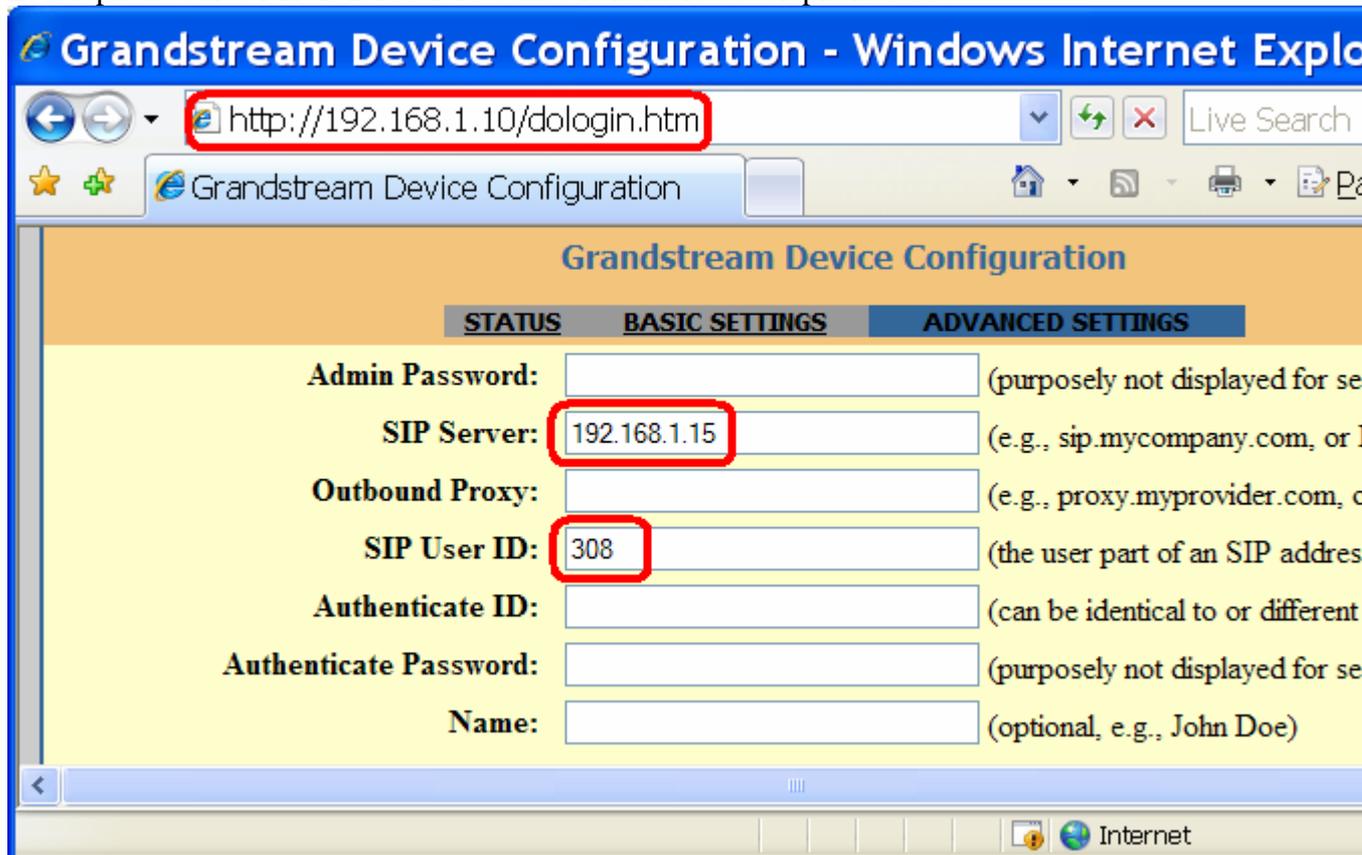
La première ligne du message contient le nom de la méthode (INVITE) Les lignes suivantes sont une liste d'en-têtes [\[rfc3261, p.10-11\]](#). Beaucoup de messages SIP sont limités par la première ligne et les en-têtes la suivant. Dans l'exemple, le message contient aussi un corps (qui n'est pas illustré). Le corps est transmis par un message SIP, similairement à un document attaché à un email, ou une page web transmise dans un message HTTP [\[rfc3261, p.12-13\]](#).

Le corps du message INVITE contient les détails de la session, comme les informations sur le codec, l'adresse IP du serveur de média ou les ports utilisés par les appareils. Ces informations n'utilisent pas le format SIP. Le corps de l'INVITE utilise un autre protocole, à savoir le Session Description Protocol (SDP), décrit dans RFC 2327 [\[txt\]](#), [\[htm\]](#).

2.3. Configuration des appareils SIP

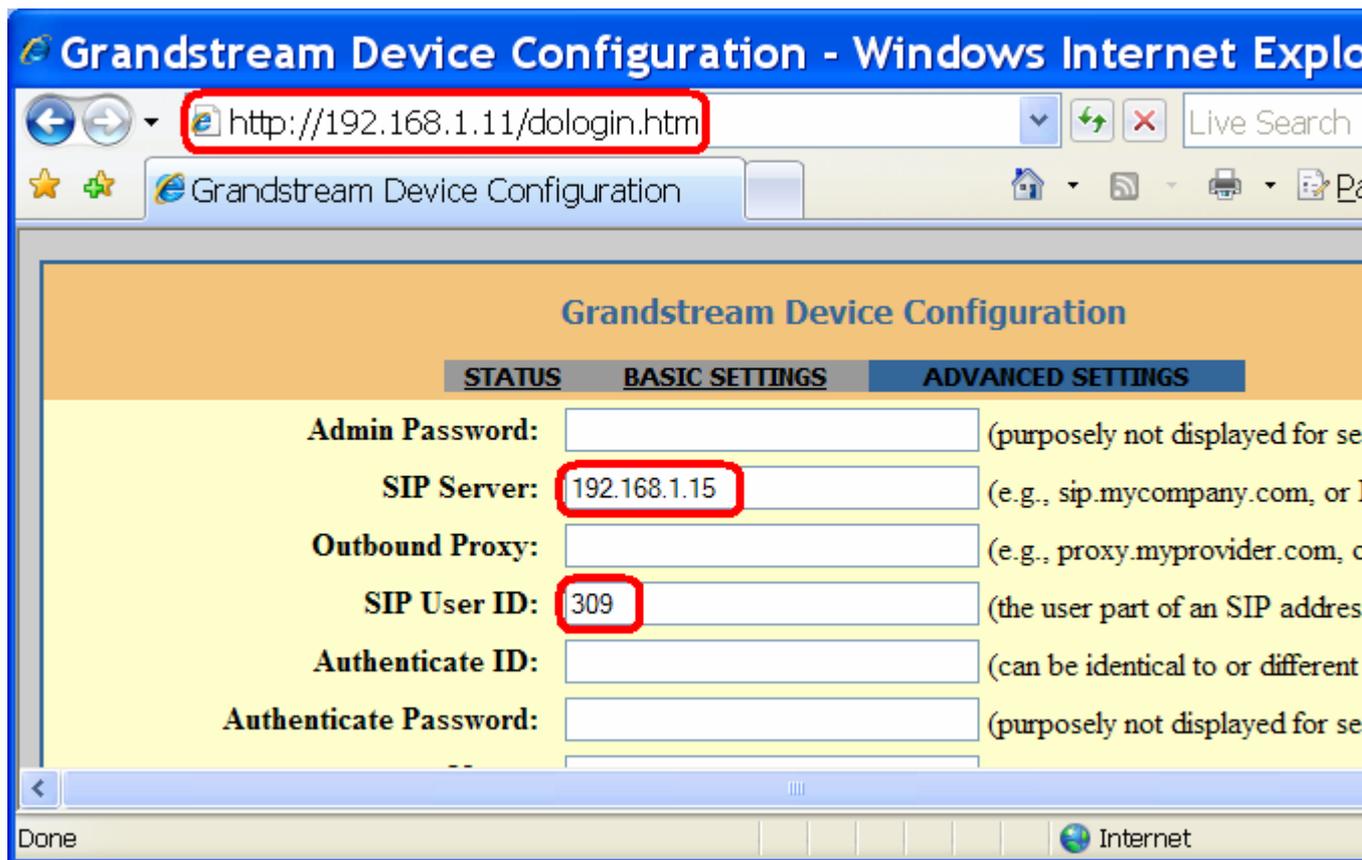
Notre configuration de test utilise deux appareils SIP Grandstream Budge Tone-100. Ils sont configurés en IP statique : 192.168.10 et 192.168.1.11, pour plus de clarté dans les explications (une configuration IP dynamique, au moyen d'un serveur DHCP, fonctionnerait aussi). Les deux appareils sont connectés au même serveur SIP, à l'adresse 192.168.1.15.

Le téléphone SIP à l'IP 192.168.1.10 utilise le numéro de téléphone 308 :



[\[htm\]](#)

Le téléphone SIP à l'IP 192.168.1.11 utilise le numéro de téléphone 309 :



[htm]

2.4. Configuration du serveur proxy SIP

Nous utilisons un proxy [OpenSER](#) à l'adresse IP 192.168.1.15. C'est une déclinaison de SIP Express Router ([SER](#)). La configuration de SER s'effectue dans un unique fichier texte. Dans sa configuration basique, SER gère les requêtes d'enregistrement (sans authentification) et permet les appels entre téléphones SIP. Le fichier de configuration fait typiquement 100-200 lignes [[ser.cfg](#)], [[openser.cfg](#)], et consiste en un script avec une syntaxe proche du C. Le fichier de configuration définit les opérations à effectuer à la réception des requêtes SIP individuelles. Les réponses peuvent aussi utiliser un fonctionnement "avec mémoire" (voir sections **Error! Reference source not found.** et 3). Les scripts de SER et OpenSER sont quasiment compatibles. Avec de petites modifications, un fichier de configuration de SER peut être utilisé dans OpenSER et vice-versa.

Pour mieux visualiser les échanges SIP, le serveur OpenSER est lancé en mode debug depuis un terminal. À la réception d'une requête SIP, son contenu entier est affiché, et nous attendons toutes les réponses à ce message [[cfg](#)]. Le traitement appliqué au paquet SIP est celui du script de configuration par défaut. Les détails du fichier de configuration sortent du cadre de ce document. Le langage utilisé dans le script est décrit dans "[SER Getting Started](#)".

Le tampon de messages est affiché en entier par la commande suivante :

```
xlog("L_INFO", "\n\n$Cbg[ Method $rm from $si ]$Cxx\n$mb$Cbg[ End
of Request ]$Cxx\n");
```

Ici, \$mb représente la totalité du tampon du message SIP, incluant le corps du message (en attachement). Le corps seul d'une requête ou réponse peut être affiché avec la pseudo variable \$rb. La méthode d'affichage xlog est définie dans le module xlog.so [[xlog ser](#)], [[xlog opener](#)], [[readme](#)] (la seule différence d'usage de xlog dans SER et OpenSER est que l'un utilise le symbole '%' pour indiquer les pseudo variables, alors que l'autre utilise '\$'). La variable \$si représente l'adresse IP de l'expéditeur du message, et la variable \$rm la méthode de requête.

Les réponses aux requêtes SIP peuvent aussi être traitées par le script SER. Dans la méthode principale de routage, on doit spécifier quelle fonction traite les réponses :

```
t_on_reply("1");
```

On doit aussi fournir le corps de la méthode qui traite les réponses :

```
onreply_route[1]
{
  xlog("L_INFO", "\n\n$Cbc[ Reply $rs ($rr) from $si concerning $rm
]$Cxx\n$mb$Cbc[ End of Reply ]$Cxx\n");
}
```

Les réponses seront traitées uniquement si les requêtes ont été précédemment relayées avec une fonction à mémoire :

```
route[1]
{
  if(!t_relay())
    sl_reply_error();
  xlog("L_INFO", "$CbxMessage is relayed; now exiting$Cxx\n");
  exit;
}
```

Dans le corps de onreply_route[1], on affiche le contenu du message de réponse (\$mb), le statut de la réponse (\$rs), la raison de la réponse (\$rr), l'adresse IP de l'expéditeur (\$si) et la méthode de requête initiale associée à laquelle la réponse est associée (\$rr). Cette association est possible grâce au traitement "à mémoire" des transactions par SER. Les réponses SIP sont corrélées aux requêtes initiales grâce aux paramètres d'identification fournis par les champs Via présents dans les en-têtes des messages SIP [[rfc3261, p.13](#)].

2.5. Messages d'établissement d'appel

Dans cet exemple un appel est effectué du téléphone SIP 308 (à l'adresse IP 192.168.1.10) au téléphone SIP 309 (à l'adresse IP 192.168.1.11). Ici le téléphone SIP destinataire (309) répond et l'appel est établi. Le serveur OpenSER – qui joue le rôle de proxy pour les deux téléphones – logue à l'écran les messages reçus (selon le fichier de configuration décrit précédemment [[cfg](#)]). En dessous se trouve la sortie du serveur pour notre appel de 308 à 309. Le message encadré par les barres vertes représente la requête INVITE ; les messages encadrés par les barres bleues sont les réponses. La requête INVITE est envoyée depuis le téléphone à l'adresse 192.168.1.10. On voit ensuite la réponse 100 (Trying) envoyée par le destinataire 192.168.1.11. Cette réponse est immédiatement suivie de 180 (Ringing), également envoyée par le destinataire. Quand l'appelant reçoit en retour 180 (Ringing), il génère la tonalité de sonnerie pour l'utilisateur. Le décrochage de l'appareil appelé est confirmé par la réponse 200 (OK) envoyée par le destinataire. Ces trois messages

constituent une **transaction SIP**, qui contient une requête INVITE avec ses trois réponses 100 (Trying) [[rfc3261, p.13](#)], 180 (Ringing), et 200 (OK).

[Method INVITE from 192.168.1.10]

```
INVITE sip:309@192.168.1.15 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.10;branch=z9hG4bK533110eb792e9593
From: <sip:308@192.168.1.15>;tag=839d16b92cebf0ae
To: <sip:309@192.168.1.15>
Contact: <sip:308@192.168.1.10>
Supported: replaces
Call-ID: 278956deb55db668@192.168.1.10
CSeq: 23290 INVITE
User-Agent: Grandstream BT110 1.0.8.33
Max-Forwards: 70
Allow: INVITE,ACK,CANCEL,BYE,NOTIFY,REFER,OPTIONS,INFO,SUBSCRIBE
Content-Type: application/sdp
Content-Length: 225
```

```
v=0
o=308 8000 8000 IN IP4 192.168.1.10
s=SIP Call
c=IN IP4 192.168.1.10
t=0 0
m=audio 5004 RTP/AVP 4 18 97
a=sendrecv
a=rtpmap:4 G723/8000
a=rtpmap:18 G729/8000
a=rtpmap:97 iLBC/8000
a=fmtp:97 mode=20
a=ptime:60
```

[End of Request]

[Reply 100 (Trying) from 192.168.1.11 concerning INVITE]

```
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 192.168.1.15;branch=z9hG4bK4599.12d426e3.0
Via: SIP/2.0/UDP 192.168.1.10;branch=z9hG4bK533110eb792e9593
From: <sip:308@192.168.1.15>;tag=839d16b92cebf0ae
To: <sip:309@192.168.1.15>
Call-ID: 278956deb55db668@192.168.1.10
CSeq: 23290 INVITE
User-Agent: Grandstream BT110 1.0.8.33
Content-Length: 0
```

[End of Reply]

[Reply 180 (Ringing) from 192.168.1.11 concerning INVITE]

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 192.168.1.15;branch=z9hG4bK4599.12d426e3.0
Via: SIP/2.0/UDP 192.168.1.10;branch=z9hG4bK533110eb792e9593
From: <sip:308@192.168.1.15>;tag=839d16b92cebf0ae
To: <sip:309@192.168.1.15>;tag=e821cb882a12d201
Call-ID: 278956deb55db668@192.168.1.10
CSeq: 23290 INVITE
User-Agent: Grandstream BT110 1.0.8.33
Content-Length: 0
```

[End of Reply]

```
[ Reply 200 (OK) from 192.168.1.11 concerning INVITE ]
```

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.1.15;branch=z9hG4bK4599.12d426e3.0
Via: SIP/2.0/UDP 192.168.1.10;branch=z9hG4bK533110eb792e9593
From: <sip:308@192.168.1.15>;tag=839d16b92cebf0ae
To: <sip:309@192.168.1.15>;tag=e821cb882a12d201
Call-ID: 278956deb55db668@192.168.1.10
CSeq: 23290 INVITE
User-Agent: Grandstream BT110 1.0.8.33
Contact: <sip:309@192.168.1.11>
Allow: INVITE,ACK,CANCEL,BYE,NOTIFY,REFER,OPTIONS,INFO,SUBSCRIBE
Content-Type: application/sdp
Supported: replaces
Content-Length: 154
```

```
v=0
o=309 8000 8000 IN IP4 192.168.1.11
s=SIP Call
c=IN IP4 192.168.1.11
t=0 0
m=audio 5004 RTP/AVP 4
a=sendrecv
a=rtpmap:4 G723/8000
a=ptime:60
```

```
[ End of Reply ]
```

[\[htm\]](#), [\[doc\]](#), [\[txt\]](#)

Tous les messages de la transaction sont transmis via le serveur proxy, et donc nous pouvons tous les voir. Dans le contexte d'une transaction SIP, les messages suivent toujours le même chemin. Le champ **Via** du header SIP est responsable de l'identification des transactions et de conserver le chemin des messages appartenant à la même transaction. La transaction est identifiée par un paramètre **branch** incorporé au champ Via [[rfc3261, p.12](#)]. Les réponses sont liées aux requêtes grâce au champ branch [[rfc3261, p.13](#)]. Dans la sortie suivante, les paramètres branch sont en gras. On peut voir qu'ils sont identiques pour les trois messages.

Le chemin des messages est conservé grâce à la pile de champs Via ajoutés par les proxys intermédiaires à l'en-tête des requêtes, lorsqu'elles traversent la chaîne de proxys (un seul est présent dans notre exemple). Le téléphone SIP s'assure que la réponse à une requête SIP contient la même pile de champs Via. Quand les réponses sont renvoyées, chaque proxy intermédiaire enlève son adresse du sommet de la pile et utilise le champ Via suivant pour déterminer ou envoyer la réponse. Ainsi, chaque proxy qui voit une requête verra aussi toutes les réponses associées à cette requête [[rfc3261, p.13-14](#)].

La transaction illustrée plus haut initie un appel téléphonique. Plusieurs autres transactions peuvent arriver pendant et à la fin de la conversation. Même si (comme expliqué) dans le cadre d'une transaction tous les messages SIP suivent le même chemin, les autres transactions liées au même appel ne vont pas nécessairement suivre le même chemin que celle qui initie l'appel. Au lieu de passer par les proxys intermédiaires, les transactions suivantes sont souvent directement transmises entre les deux téléphones SIP (voir le premier exemple de RFC3261 [[rfc3261, p.15](#)]). Cela arrive car pendant la première transaction INVITE/200 (OK), les nœuds finaux

apprennent leurs adresses respectives depuis les champs Contact des en-têtes [[rfc3261, p.12](#)].

L'exemple de cette section correspond aussi à un scénario similaire au premier exemple de RFC3261 [[rfc3261, p.10-11](#)]. Dès que 100 (OK) est transmis pour le premier INVITE (c-à-d que l'appel est établi), plus aucun message de signalisation ne passe par le proxy [[rfc3261, p.15](#)]. Toutes les transactions SIP suivantes passent directement entre les téléphones SIP, contournant le proxy. Ainsi on ne voit plus les autres messages de l'appel (qui contiennent au moins ACK, BYE et la réponse 200 (OK) du BYE).

Dans la requête INVITE et la réponse 200 (OK) de la sortie imprimée, on voit aussi le corps des messages SDP attachés [[rfc3261, p.12-13](#)]. Par l'échange de ces messages SDP, les téléphones SIP négocient les paramètres de la future session média. En particulier, chaque appareil informe l'autre sur les adresses et ports disponibles pour le flux de média (ici 192.168.1.10:5004 et 192.168.1.11:5004) et négocient le codec G723 pour le média.

2.6. Appel annulé

L'appelant peut raccrocher avant que l'appelé ne réponde. Cela ajoute des messages supplémentaires dans la transaction d'établissement de l'appel. La sortie suivante correspond à ce scénario. D'abord le téléphone 308 (à l'IP 192.168.1.10) compose le numéro 309, et raccroche sans attendre la réponse de 309.

```
[ Method INVITE from 192.168.1.10 ]
INVITE sip:309@192.168.1.15 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.10;branch=z9hG4bKdfda7b9079412bd5
From: <sip:308@192.168.1.15>;tag=8b2723dc35649705
To: <sip:309@192.168.1.15>
Contact: <sip:308@192.168.1.10>
Supported: replaces
Call-ID: 64f567bcc5d8f80c@192.168.1.10
CSeq: 14107 INVITE
User-Agent: Grandstream BT110 1.0.8.33
Max-Forwards: 70
Allow: INVITE,ACK,CANCEL,BYE,NOTIFY,REFER,OPTIONS,INFO,SUBSCRIBE
Content-Type: application/sdp
Content-Length: 225

v=0
o=308 8000 8000 IN IP4 192.168.1.10
s=SIP Call
c=IN IP4 192.168.1.10
t=0 0
m=audio 5004 RTP/AVP 4 18 97
a=sendrecv
a=rtpmap:4 G723/8000
a=rtpmap:18 G729/8000
a=rtpmap:97 iLBC/8000
a=fmtp:97 mode=20
a=ptime:60
[ End of Request ]
```

[Reply 100 (Trying) from 192.168.1.11 concerning INVITE]

SIP/2.0 100 Trying
Via: SIP/2.0/UDP 192.168.1.15;branch=z9hG4bK5647.03eb25b7.0
Via: SIP/2.0/UDP 192.168.1.10;branch=z9hG4bKdfda7b9079412bd5
From: <sip:308@192.168.1.15>;tag=8b2723dc35649705
To: <sip:309@192.168.1.15>
Call-ID: 64f567bcc5d8f80c@192.168.1.10
CSeq: 14107 INVITE
User-Agent: Grandstream BT110 1.0.8.33
Content-Length: 0

[End of Reply]

[Reply 180 (Ringing) from 192.168.1.11 concerning INVITE]

SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 192.168.1.15;branch=z9hG4bK5647.03eb25b7.0
Via: SIP/2.0/UDP 192.168.1.10;branch=z9hG4bKdfda7b9079412bd5
From: <sip:308@192.168.1.15>;tag=8b2723dc35649705
To: <sip:309@192.168.1.15>;tag=76adf65f887d5f3f
Call-ID: 64f567bcc5d8f80c@192.168.1.10
CSeq: 14107 INVITE
User-Agent: Grandstream BT110 1.0.8.33
Content-Length: 0

[End of Reply]

[Method CANCEL from 192.168.1.10]

CANCEL sip:309@192.168.1.15 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.10;branch=z9hG4bKdfda7b9079412bd5
From: <sip:308@192.168.1.15>;tag=8b2723dc35649705
To: <sip:309@192.168.1.15>
Supported: replaces
Call-ID: 64f567bcc5d8f80c@192.168.1.10
CSeq: 14107 CANCEL
User-Agent: Grandstream BT110 1.0.8.33
Max-Forwards: 70
Allow: INVITE,ACK,CANCEL,BYE,NOTIFY,REFER,OPTIONS,INFO,SUBSCRIBE
Content-Length: 0

[End of Request]

[Reply 200 (OK) from 192.168.1.11 concerning CANCEL]

SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.1.15;branch=z9hG4bK5647.03eb25b7.0
Via: SIP/2.0/UDP 192.168.1.10;branch=z9hG4bKdfda7b9079412bd5
From: <sip:308@192.168.1.15>;tag=8b2723dc35649705
To: <sip:309@192.168.1.15>;tag=76adf65f887d5f3f
Call-ID: 64f567bcc5d8f80c@192.168.1.10
CSeq: 14107 CANCEL
User-Agent: Grandstream BT110 1.0.8.33
Contact: <sip:309@192.168.1.11>
Allow: INVITE,ACK,CANCEL,BYE,NOTIFY,REFER,OPTIONS,INFO,SUBSCRIBE
Supported: replaces
Content-Length: 0

[End of Reply]

0(22535)

[Reply 487 (Request Cancelled) from 192.168.1.11 concerning INVITE]

SIP/2.0 487 Request Cancelled
Via: SIP/2.0/UDP 192.168.1.15;branch=z9hG4bK5647.03eb25b7.0
Via: SIP/2.0/UDP 192.168.1.10;branch=z9hG4bKdfda7b9079412bd5
From: <sip:308@192.168.1.15>;tag=8b2723dc35649705
To: <sip:309@192.168.1.15>;tag=76adf65f887d5f3f
Call-ID: 64f567bcc5d8f80c@192.168.1.10
CSeq: 14107 INVITE
User-Agent: Grandstream BT110 1.0.8.33
Content-Length: 0

[End of Reply]

0(22535)

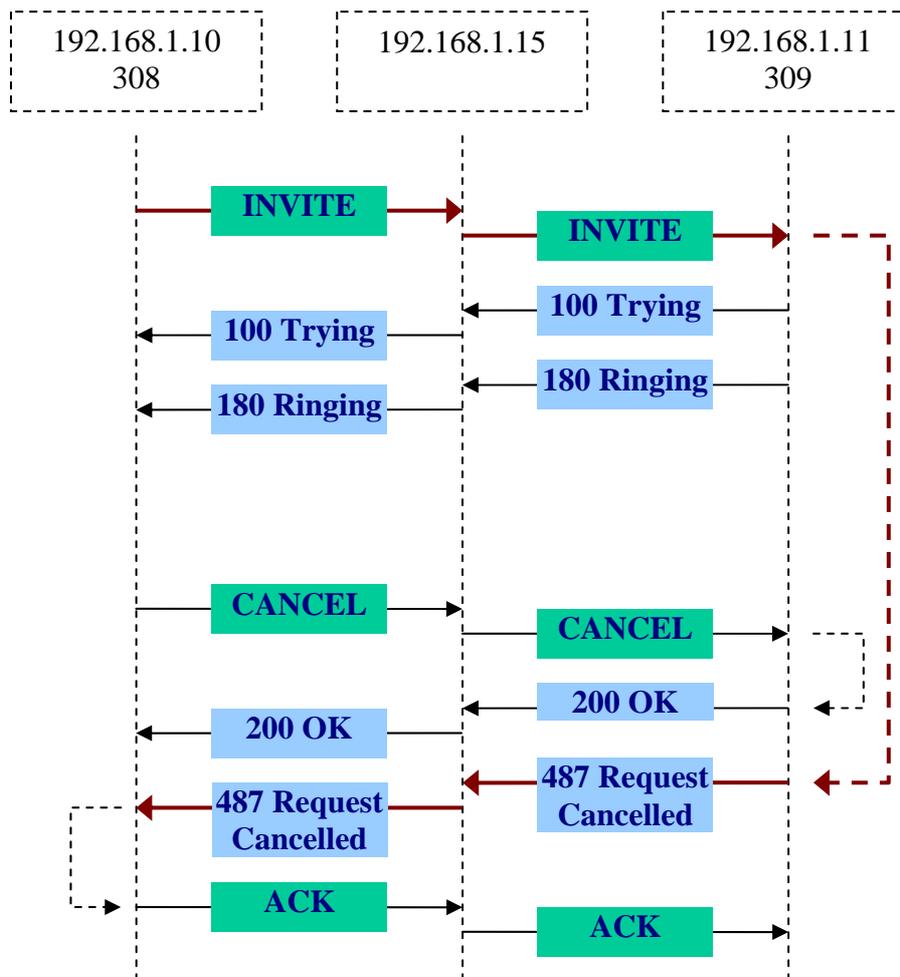
[Method ACK from 192.168.1.10]

ACK sip:309@192.168.1.15 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.10;branch=z9hG4bKdfda7b9079412bd5
From: <sip:308@192.168.1.15>;tag=8b2723dc35649705
To: <sip:309@192.168.1.15>;tag=76adf65f887d5f3f
Contact: <sip:308@192.168.1.10>
Call-ID: 64f567bcc5d8f80c@192.168.1.10
CSeq: 14107 ACK
User-Agent: Grandstream BT110 1.0.8.33
Max-Forwards: 70
Allow: INVITE,ACK,CANCEL,BYE,NOTIFY,REFER,OPTIONS,INFO,SUBSCRIBE
Content-Length: 0

[End of Request]

[\[htm\]](#), [\[doc\]](#), [\[txt\]](#)

Les sept messages correspondent à la même transaction, Cette transaction essaie d'établir l'appel, mais ensuite annule la tentative. Voici le diagramme des messages échangés :



Les en-têtes affichés dans la sortie contiennent deux paramètres branch. Le premier (branch=z9hG4bKdfda7b9079412bd5) se trouve dans le champ Via ajouté par le téléphone SIP appelé 192.168.1.10. Le second (branch=z9hG4bK5647.03eb25b7.0) se trouve dans le champ Via ajouté par le proxy SIP. La transaction peut être identifiée par n'importe lequel de ces deux paramètres branch. Notons que dans ce scénario – où l'appel est annulé avant d'être établi – le message ACK est transmis dans le cadre de la même transaction (il a le même paramètre branch), alors que dans le cas où l'appel est répondu (voir la sortie de la section **Error! Reference source not found.**), le message ACK est envoyé avec un paramètre branch différent et n'appartient donc pas à la transaction de la requête INVITE.

Quand le proxy SIP 192.168.1.15 reçoit une requête INVITE et voit "branch=z9hG4bKdfda7b9079412bd5" pour la première fois (le paramètre branch étant ajouté par le téléphone SIP appelant 192.168.1.10), le proxy crée un numéro de transaction interne z9hG4bK5647.03eb25b7.0 associé à tous les messages attendus suivants de la transaction. Le numéro de transaction interne créé par le proxy et le paramètre branch sont synonymes et identifient la même transaction. Quand le proxy SIP reçoit de nouvelles requêtes provenant de 192.168.1.10 avec le même paramètre branch (fourni par le téléphone appelant), le proxy peut identifier la transaction sortante et va associer la transaction interne déjà créée avec ces messages. De plus, le proxy ajoute son numéro de transaction interne dans le champ Via ajouté aux en-têtes de messages.

La sortie de cet exemple montre le contenu des messages SIP à leur arrivée, avant que le champ Via ne soit inclus dans les requêtes. Les réponses SIP sont transférées en retour du téléphone SIP 192.168.1.11 au proxy 192.168.1.15, et elles contiennent les en-têtes Via du téléphone appelant et du proxy lui-même (insérés dans la requête SIP pendant son transfert vers le téléphone appelant). Le proxy à mémoire associe le message de réponse à la transaction en cours en utilisant soit son propre paramètre branch ou celui du téléphone appelant. Par exemple, il est possible d'accéder à la méthode de requête (par la pseudo variable \$rm) pendant le traitement des messages de réponse, grâce à l'association entre la réponse et la transaction correspondante. Avant de relayer la réponse au téléphone source, le proxy enlève son propre champ Via de l'en-tête de la réponse [[rfc3261, p.13-14](#)]. La sortie affichée montre les messages de réponse à leur arrivée sur le proxy, donc le champ Via du header n'est pas encore enlevé.

Le téléphone d'origine apprend l'information de contact direct vers le téléphone SIP de destination depuis l'en-tête Contact de la réponse 200 (OK) au message CANCEL (en gras dans la sortie). Toutefois, le message ACK n'est pas envoyé directement. Le ACK contient le même numéro de branche, et est donc dans le contexte de la même transaction. Ainsi, le ACK suit le même chemin via le serveur proxy.

Comme la tentative d'appel est annulée, dans la transaction présentée seul le message INVITE contient un attachement SDP. Il n'y a pas besoin d'attachement SDP dans la réponse 487 (Request Cancelled).

2.7. Configuration de SER sans mémoire (stateless)

En utilisant les fonctions sans mémoire de SER, le proxy transmet simplement les messages sans garder en mémoire les informations de transaction. Pour du forwarding sans mémoire, on utilise la méthode forward() (t_relay() pour un forwarding avec mémoire) :

```
route[1]
{
  if(!forward())
    sl_reply_error();
  xlog("L_INFO", "$CbxMessage is relayed; now exiting$Cxx\n");
  exit;
}
```

Avec le traitement sans mémoire, les messages sont traités sans contexte et les réponses ne peuvent être associées aux requêtes [[SER Getting Started, p.9](#)]. Cela signifie qu'avant de relayer une requête SIP, le proxy ne stocke pas en mémoire les paramètres de branch de la requête SIP pour traiter les réponses correspondantes.

Toujours avec les méthodes stateless[[cfg](#)], pendant l'essai d'appel annulé, seuls les messages INVITE et CANCEL seront vus et logués par le script SER (voir la sortie ci-dessous). Tous les autres messages associés à la transaction ne seront pas traités :

```
[ Method INVITE from 192.168.1.10 ]
INVITE sip:309@192.168.1.15 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.10;branch=z9hG4bKef1e1f8da5c0298d
From: <sip:308@192.168.1.15>;tag=0434481ac70e589b
To: <sip:309@192.168.1.15>
```

```
Contact: <sip:308@192.168.1.10>
Supported: replaces
Call-ID: f146448b7830985b@192.168.1.10
CSeq: 24573 INVITE
User-Agent: Grandstream BT110 1.0.8.33
Max-Forwards: 70
Allow: INVITE,ACK,CANCEL,BYE,NOTIFY,REFER,OPTIONS,INFO,SUBSCRIBE
Content-Type: application/sdp
Content-Length: 225

v=0
o=308 8000 8000 IN IP4 192.168.1.10
s=SIP Call
c=IN IP4 192.168.1.10
t=0 0
m=audio 5004 RTP/AVP 4 18 97
a=sendrecv
a=rtpmap:4 G723/8000
a=rtpmap:18 G729/8000
a=rtpmap:97 iLBC/8000
a=fmtp:97 mode=20
a=ptime:60
[ End of Request ]

[ Method CANCEL from 192.168.1.10 ]
CANCEL sip:309@192.168.1.15 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.10;branch=z9hG4bKef1e1f8da5c0298d
From: <sip:308@192.168.1.15>;tag=0434481ac70e589b
To: <sip:309@192.168.1.15>
Supported: replaces
Call-ID: f146448b7830985b@192.168.1.10
CSeq: 24573 CANCEL
User-Agent: Grandstream BT110 1.0.8.33
Max-Forwards: 70
Allow: INVITE,ACK,CANCEL,BYE,NOTIFY,REFER,OPTIONS,INFO,SUBSCRIBE
Content-Length: 0
[ End of Request ]
```

[\[htm\]](#), [\[doc\]](#), [\[txt\]](#)

3. Le proxy SIP traite l'établissement de l'appel et tous les messages de signalisation suivants

Dans la section **Error! Reference source not found.**, nous avons vu l'établissement d'un appel entre deux téléphones SIP. Dans l'exemple correspondant, le proxy SIP participe à l'établissement de l'appel mais ne voit aucun des messages SIP après cette transaction. Les en-têtes SIP donnent aux proxys intermédiaires la possibilité de rester dans le chemin de la signalisation pour la suite de l'appel. Les messages de signalisation entre les destinataires finaux peuvent être entendus durant toute la durée de la session SIP. Si le proxy veut rester dans le chemin des messages SIP après la transaction d'établissement de l'appel (i.e. l'INVITE et les messages associés), il devrait ajouter à l'INVITE un en-tête de routage, appelé **Record-Route**, qui contient

le nom d'hôte ou l'adresse IP du proxy. Cette information est reçue par les téléphones SIP appelé et appelant grâce au header Record-Route, qui est transmis dans les réponses 180 (Ringing) et 200 (OK). Ils sont indiqués en gras dans l'exemple de sortie suivant. L'information Record-Route est stockée pour toute la durée du dialogue. Le serveur proxy va ensuite recevoir aussi les messages ACK et BYE / 200 (OK). Chaque proxy peut décider indépendamment de recevoir les messages suivants, et ces messages vont passer par tous les proxys qui demandent à les recevoir [[rfc3261, p.16](#)].

Le champ Record-Route est ajouté au message par les lignes suivantes du fichier de configuration [cfg] :

```
if(method!="REGISTER") {
    xlog("L_INFO","$CbxAdding the Route header$Cxx\n");
    record_route();
}

if(loose_route()) {
    xlog("L_INFO","$CbxLoose Route$Cxx\n");
    route(1);
}
```

La sortie ci-dessous montre que notre proxy a bien reçu tous les messages de signalisation, de l'établissement de l'appel à la fin de la communication.

Quand un téléphone SIP reçoit la pile de headers **Record-Route** (ajoutés par chaque proxy intermédiaire désirant rester dans le dialogue), il copie cette pile dans une nouvelle pile de champs **Route**. Ces champs Route indiquent le chemin à suivre pour les messages sortants envoyés par le téléphone. L'appareil va ajouter les en-têtes Route à toutes les requêtes transmises dans le cadre de la session SIP courante (les champs Route sont indiqués en gras dans la sortie suivante).

Quand le téléphone appelé reçoit une requête avec une pile de headers Record-Route, il copie les mêmes champs Record-Route dans le message de réponse. Ainsi, le téléphone SIP appelant va aussi recevoir cette pile de champs Record-Route. Le téléphone appelant crée aussi une pile de headers Route, mais cette pile est inversée (comme la séquence de proxys va être traversée dans l'ordre inverse).

Le serveur SER ne mémorise pas l'état du dialogue SIP (il ne voit pas les appels sortants). C'est au téléphone SIP de s'assurer que dans le cadre d'un appel, la pile de headers Route soit ajoutée dans chaque message transmis (assurant ainsi que les serveurs SIP intermédiaires reçoivent bien tous les messages se rapportant à l'appel en cours).

```
[ Method INVITE from 192.168.1.10 ]
INVITE sip:309@192.168.1.15 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.10;branch=z9hG4bKf3c60221ce03445c
From: <sip:308@192.168.1.15>;tag=fce371520693b722
To: <sip:309@192.168.1.15>
Contact: <sip:308@192.168.1.10>
Supported: replaces
Call-ID: 8c80d06175b1eb80@192.168.1.10
CSeq: 28826 INVITE
User-Agent: Grandstream BT110 1.0.8.33
Max-Forwards: 70
```

Allow: INVITE,ACK,CANCEL,BYE,NOTIFY,REFER,OPTIONS,INFO,SUBSCRIBE
Content-Type: application/sdp
Content-Length: 225

v=0
o=308 8000 8000 IN IP4 192.168.1.10
s=SIP Call
c=IN IP4 192.168.1.10
t=0 0
m=audio 5004 RTP/AVP 4 18 97
a=sendrecv
a=rtpmap:4 G723/8000
a=rtpmap:18 G729/8000
a=rtpmap:97 iLBC/8000
a=fmtp:97 mode=20
a=ptime:60
{end}

[End of Request]

[Reply 100 (Trying) from 192.168.1.11 concerning INVITE]

SIP/2.0 100 Trying
Via: SIP/2.0/UDP 192.168.1.15;branch=z9hG4bKb992.ec6bce33.0
Via: SIP/2.0/UDP 192.168.1.10;branch=z9hG4bKf3c60221ce03445c
From: <sip:308@192.168.1.15>;tag=fce371520693b722
To: <sip:309@192.168.1.15>
Call-ID: 8c80d06175bleb80@192.168.1.10
CSeq: 28826 INVITE
User-Agent: Grandstream BT110 1.0.8.33
Content-Length: 0

[End of Reply]

[Reply 180 (Ringing) from 192.168.1.11 concerning INVITE]

SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 192.168.1.15;branch=z9hG4bKb992.ec6bce33.0
Via: SIP/2.0/UDP 192.168.1.10;branch=z9hG4bKf3c60221ce03445c
Record-Route: <sip:192.168.1.15;lr=on;ftag=fce371520693b722>
From: <sip:308@192.168.1.15>;tag=fce371520693b722
To: <sip:309@192.168.1.15>;tag=352ec99f5ffd23cf
Call-ID: 8c80d06175bleb80@192.168.1.10
CSeq: 28826 INVITE
User-Agent: Grandstream BT110 1.0.8.33
Content-Length: 0

[End of Reply]

[Reply 200 (OK) from 192.168.1.11 concerning INVITE]

SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.1.15;branch=z9hG4bKb992.ec6bce33.0
Via: SIP/2.0/UDP 192.168.1.10;branch=z9hG4bKf3c60221ce03445c
Record-Route: <sip:192.168.1.15;lr=on;ftag=fce371520693b722>
From: <sip:308@192.168.1.15>;tag=fce371520693b722
To: <sip:309@192.168.1.15>;tag=352ec99f5ffd23cf
Call-ID: 8c80d06175bleb80@192.168.1.10
CSeq: 28826 INVITE
User-Agent: Grandstream BT110 1.0.8.33
Contact: <sip:309@192.168.1.11>
Allow: INVITE,ACK,CANCEL,BYE,NOTIFY,REFER,OPTIONS,INFO,SUBSCRIBE

Content-Type: application/sdp
Supported: replaces
Content-Length: 154

v=0
o=309 8000 8000 IN IP4 192.168.1.11
s=SIP Call
c=IN IP4 192.168.1.11
t=0 0
m=audio 5004 RTP/AVP 4
a=sendrecv
a=rtpmap:4 G723/8000
a=ptime:60
{end}

[End of Reply]

[Method ACK from 192.168.1.10]

ACK sip:309@192.168.1.11 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.10;branch=z9hG4bK9678e44f4ea235df
Route: <sip:192.168.1.15;lr=on;ftag=fce371520693b722>
From: <sip:308@192.168.1.15>;tag=fce371520693b722
To: <sip:309@192.168.1.15>;tag=352ec99f5ffd23cf
Contact: <sip:308@192.168.1.10>
Call-ID: 8c80d06175b1eb80@192.168.1.10
CSeq: 28826 ACK
User-Agent: Grandstream BT110 1.0.8.33
Max-Forwards: 70
Allow: INVITE,ACK,CANCEL,BYE,NOTIFY,REFER,OPTIONS,INFO,SUBSCRIBE
Content-Length: 0

[End of Request]

[Method INFO from 192.168.1.10]

INFO sip:309@192.168.1.11 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.10;branch=z9hG4bK93bc61c18d96eb8b
Route: <sip:192.168.1.15;lr=on;ftag=fce371520693b722>
From: <sip:308@192.168.1.15>;tag=fce371520693b722
To: <sip:309@192.168.1.15>;tag=352ec99f5ffd23cf
Contact: <sip:308@192.168.1.10>
Supported: replaces
Call-ID: 8c80d06175b1eb80@192.168.1.10
CSeq: 28827 INFO
User-Agent: Grandstream BT110 1.0.8.33
Max-Forwards: 70
Allow: INVITE,ACK,CANCEL,BYE,NOTIFY,REFER,OPTIONS,INFO,SUBSCRIBE
Content-Type: application/dtmf-relay
Content-Length: 23

Signal=2
Duration=4800 {end}

[End of Request]

[Reply 200 (OK) from 192.168.1.11 concerning INFO]

SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.1.15;branch=z9hG4bKc992.e100efc6.0
Via: SIP/2.0/UDP 192.168.1.10;branch=z9hG4bK93bc61c18d96eb8b
Record-Route: <sip:192.168.1.15;lr=on;ftag=fce371520693b722>
From: <sip:308@192.168.1.15>;tag=fce371520693b722

To: <sip:309@192.168.1.15>;tag=352ec99f5ffd23cf
Call-ID: 8c80d06175b1eb80@192.168.1.10
CSeq: 28827 INFO
User-Agent: Grandstream BT110 1.0.8.33
Contact: <sip:309@192.168.1.11>
Allow: INVITE,ACK,CANCEL,BYE,NOTIFY,REFER,OPTIONS,INFO,SUBSCRIBE
Supported: replaces
Content-Length: 0

[End of Reply]

[Method INFO from 192.168.1.11]

INFO sip:308@192.168.1.10 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.11;branch=z9hG4bKd51a30c261b5691e
Route: <sip:192.168.1.15;lr=on;ftag=fce371520693b722>
From: <sip:309@192.168.1.15>;tag=352ec99f5ffd23cf
To: <sip:308@192.168.1.15>;tag=fce371520693b722
Contact: <sip:309@192.168.1.11>
Supported: replaces
Call-ID: 8c80d06175b1eb80@192.168.1.10
CSeq: 52519 INFO
User-Agent: Grandstream BT110 1.0.8.33
Max-Forwards: 70
Allow: INVITE,ACK,CANCEL,BYE,NOTIFY,REFER,OPTIONS,INFO,SUBSCRIBE
Content-Type: application/dtmf-relay
Content-Length: 22

Signal=3

Duration=480{end}

[End of Request]

[Reply 200 (OK) from 192.168.1.10 concerning INFO]

SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.1.15;branch=z9hG4bKb593.5d620aa5.0
Via: SIP/2.0/UDP 192.168.1.11;branch=z9hG4bKd51a30c261b5691e
Record-Route: <sip:192.168.1.15;lr=on;ftag=352ec99f5ffd23cf>
From: <sip:309@192.168.1.15>;tag=352ec99f5ffd23cf
To: <sip:308@192.168.1.15>;tag=fce371520693b722
Call-ID: 8c80d06175b1eb80@192.168.1.10
CSeq: 52519 INFO
User-Agent: Grandstream BT110 1.0.8.33
Contact: <sip:308@192.168.1.10>
Allow: INVITE,ACK,CANCEL,BYE,NOTIFY,REFER,OPTIONS,INFO,SUBSCRIBE
Supported: replaces
Content-Length: 0

[End of Reply]

[Method BYE from 192.168.1.11]

BYE sip:308@192.168.1.10 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.11;branch=z9hG4bKd0b04fbb081eb9ab
Route: <sip:192.168.1.15;lr=on;ftag=352ec99f5ffd23cf>
From: <sip:309@192.168.1.15>;tag=352ec99f5ffd23cf
To: <sip:308@192.168.1.15>;tag=fce371520693b722
Supported: replaces
Call-ID: 8c80d06175b1eb80@192.168.1.10
CSeq: 52520 BYE
User-Agent: Grandstream BT110 1.0.8.33

```
Max-Forwards: 70
Allow: INVITE,ACK,CANCEL,BYE,NOTIFY,REFER,OPTIONS,INFO,SUBSCRIBE
Content-Length: 0
```

[End of Request]

[Reply 200 (OK) from 192.168.1.10 concerning BYE]

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.1.15;branch=z9hG4bK0593.583e3506.0
Via: SIP/2.0/UDP 192.168.1.11;branch=z9hG4bKd0b04fbb081eb9ab
Record-Route: <sip:192.168.1.15;lr=on;ftag=352ec99f5ffd23cf>
From: <sip:309@192.168.1.15>;tag=352ec99f5ffd23cf
To: <sip:308@192.168.1.15>;tag=fce371520693b722
Call-ID: 8c80d06175b1eb80@192.168.1.10
CSeq: 52520 BYE
User-Agent: Grandstream BT110 1.0.8.33
Contact: <sip:308@192.168.1.10>
Allow: INVITE,ACK,CANCEL,BYE,NOTIFY,REFER,OPTIONS,INFO,SUBSCRIBE
Supported: replaces
Content-Length: 0
```

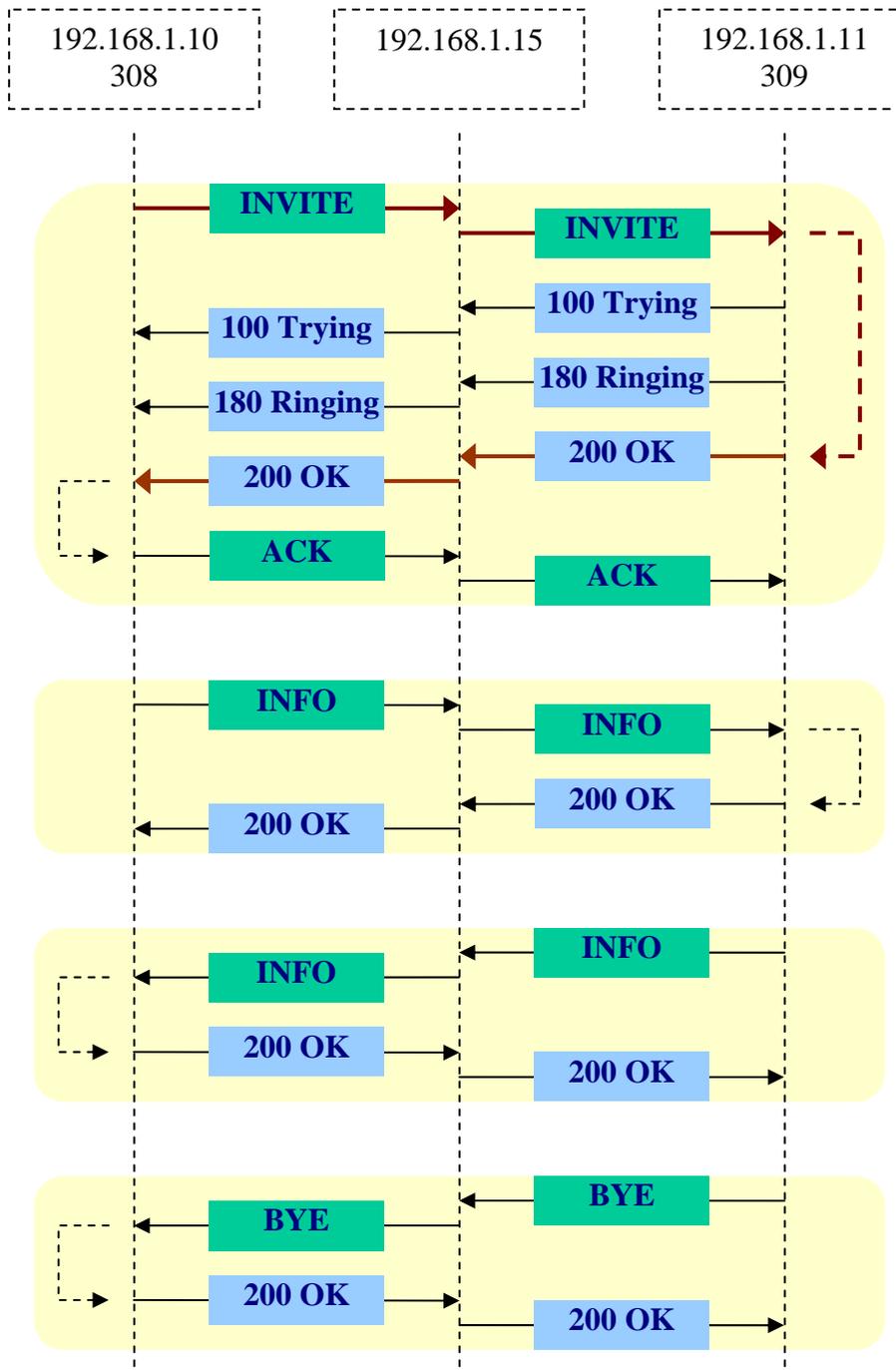
[End of Reply]

[\[htm\]](#), [\[doc\]](#), [\[txt\]](#)

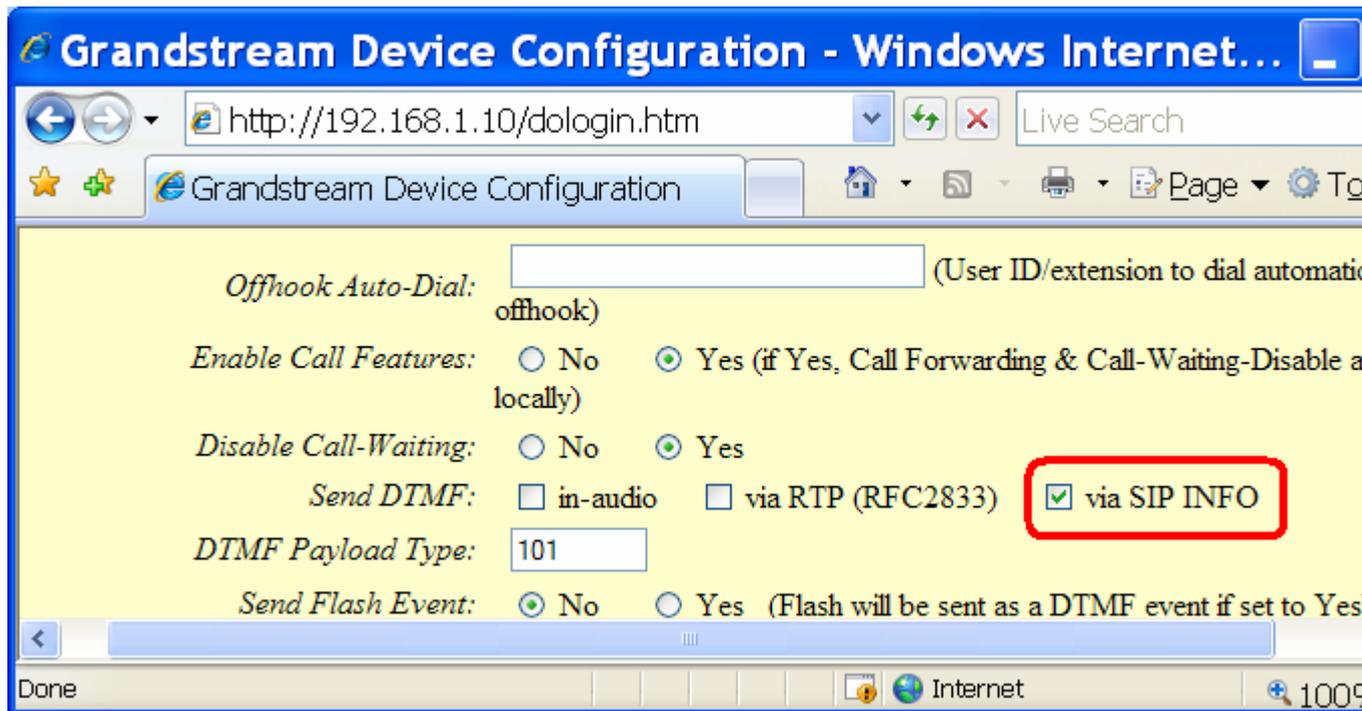
Même si le serveur SIP n'identifie pas les conversations téléphoniques, il peut enregistrer chaque message reçu dans une base de données. Une application externe peut donc réunir les messages faisant partie d'une même conversation. Une conversation peut être identifiée par le tag du téléphone appelant (tag=fce371520693b722) [\[rfc3261, p.12\]](#), celui du téléphone appelé (tag=352ec99f5ffd23cf) [\[rfc3261, p.14-15\]](#), et par le champ Call-Id (8c80d06175b1eb80@192.168.1.10). Ces tags ainsi que le champ Call-Id sont générés par les téléphones SIP en bout de chaîne, et sont gardés constants durant la conversation téléphonique. Les tags sont des chaînes de caractères aléatoires, et le Call-Id est généré en combinant une chaîne aléatoire et l'adresse IP (ou nom d'hôte) de l'appareil appelant. La combinaison des tags To et From, et du champ Call-Id identifie une relation de pair à pair entre deux téléphones SIP, et est appelée **dialogue** [\[rfc3261, p.12\]](#).

Comme un appel téléphonique (p. ex. un dialogue SIP) consiste en plusieurs transactions, et étant donné que SER ne conserve pas d'informations sur les transactions au cours du même appel, SER ne peut pas savoir si l'appel est entrant ou sortant, ni connaître la durée d'un appel (en cours ou terminé) ou le terminer. Toutefois, SER peut loguer les messages à la réception (INVITE, ACK et BYE sont stockés avec le Call-Id correspondant). Une application de facturation peut donc réunir les messages INVITE et BYE, et ainsi calculer la durée de l'appel [\[SER Getting Started, p.9\]](#).

Le dialogue affiché en exemple peut être illustré par le diagramme suivant :



Pendant la conversation, on voit deux transactions comportant un message INFO. Elles correspondent à la transmission de signaux DTMF via les messages SIP. La première indique que la touche "2" a été pressée sur le téléphone appelant pendant 4800 millisecondes. Pour s'assurer que les messages SIP INFO soient transmis, on doit spécifier dans la configuration que les informations DTMF doivent être transmises dans les messages SIP :



[bmp]

D'après l'affichage de sortie, on peut conclure que l'appel a été établi avec succès, comme le message ACK (branch=z9hG4bK9678e44f4ea235df) n'appartient plus à la transaction du message INVITE (branch=z9hG4bKf3c60221ce03445c). C'est la raison pour laquelle dans la section **Error! Reference source not found.** – sans utiliser les champs Record-Route – le message ACK ne passait pas par le serveur proxy.

Il est important de rappeler que le header Record-Route ne donne d'informations qu'aux téléphones SIP en bout de chaîne, pour ajouter les champs Route aux requêtes durant toute la durée de l'appel. Ainsi, ce sont les téléphones (destinataires finaux) qui sont chargés de vérifier que les champs Route soient ajoutés à tous les messages associés à l'appel en cours. Ils sont donc au courant de l'appel en cours, ce qui n'est pas le cas du proxy.

4. Autres expérimentations pour comprendre le protocole SIP

[Paramètres STUN des appareils SIP](#)

[Créer et envoyer des messages SIP INVITE et CANCEL](#)

[Appels directs entre deux téléphones SIP sans passer par un proxy SIP](#)

[Comprendre les échanges SIP par l'expérimentation](#)

Ce document [[htm](#)], [[pdf](#)], [[doc](#)], la page web entière [[zip](#)]

5. Glossaire

FQDN	Fully Qualified Domain Name
SDP	Session Description Protocol
HTTP	Hyper Text Transfer Protocol
RFC	Request for Comments
SIP	Session Initiation Protocol
SER	SIP Express Router
RTP	Real-time Transport Protocol
UA	User Agent
ACK	Acknowledgement
DTMF	Dual Tone Multi Frequency
URI	Universal Resource Identifier
DHCP	Dynamic Host Configuration Protocol

6. Liens en rapport

<http://openser.org/docs/pseudo-variables-1.1.x.html>, pseudo variables (openser)
<http://www iptel.org/ser/doc/modules/xlog>, pseudo variables pour xlog (ser)
<http://www iptel.org/ser/doc/gettingstarted>, documentation de démarrage (ser)
<ftp://siprouter.onsip.org/pub/gettingstarted/configs/>, fichiers de configuration simples
<http://www.faqs.org/rfcs/rfc3261.html>, SIP: Session Initiation Protocol
<http://www.faqs.org/rfcs/rfc2327.html>, SDP: Session Description Protocol
<http://www iptel.org/ser/>, page web de Sip Express Router
<http://www.openser.org/>, page web de OpenSER
<http://www.openser.org/dokuwiki/doku.php/core-cookbook:1.1.x>, mots clés, valeurs et paramètres principaux de la configuration de SER
<http://www.openser.org/docs/modules/1.2.x/textops.html>, expressions régulières avec SER
<http://mit.edu/sip/sip.edu/ser.shtml>, quelques exemples utilisant SER
http://www iptel.org/how_can_i_dynamically_change_the_invite_timer_fr_invite, modifier (dynamiquement) le délai de timeout