

# RFC 3261 : SIP: Session Initiation Protocol

Stéphane Bortzmeyer  
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 8 avril 2006. Dernière mise à jour le 7 juin 2008

Date de publication du RFC : Juin 2002

<https://www.bortzmeyer.org/3261.html>

---

Le protocole SIP, que spécifie ce RFC, a bien des usages mais le plus connu est celui de permettre la téléphonie sur IP. SIP est une tentative de protocole standard, dans un monde où les solutions spécifiques et fermées sont les plus médiatisées.

SIP permet d'établir, modifier et terminer des sessions multimédia (en pratique, surtout de la téléphonie). Il est important de noter que SIP ne transporte pas lui-même les données multimédia : son seul travail est de gérer les sessions, de permettre à deux entités, les "*User Agent*", de se reconstruire et d'ouvrir la session où vont passer les données. Celles-ci seront transportées par un autre protocole comme RTP (RFC 3550<sup>1</sup>).

Avec près de 650 000 signes, soit 270 pages, notre RFC est un des plus gros. C'est que le multimédia est chose complexe et qu'il y a énormément de détails à préciser. Mais le principe est relativement simple : l'"*User Agent*" "client" contacte l'"*User Agent*" "serveur", celle-ci "sonne" l'utilisateur et, s'il est d'accord, il "décroche" et son "*User Agent*" accepte alors la session. En pratique, les "*User Agent*" sont des téléphones SIP ou bien des "*softphones*", des applications comme WengoPhone (il est possible que l'objet téléphone disparaisse dans le futur, la téléphonie devenant juste une application informatique parmi d'autres).

Pour ce dialogue, SIP utilise des commandes texte, comme SMTP ou HTTP. On les voit ici lorsqu'on utilise la commande sipsak <<http://sipsak.org/>>, très pratique pour tester SIP :

---

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc3550.txt>

```

% sudo sipsak -v -v -s sip:beatrice@durand.fr
[...]

** request **
OPTIONS sip:beatrice@durand.fr SIP/2.0
Via: SIP/2.0/UDP foobar.bortzmeyer.org:33196;rport
From: sip:sipsak@foobar.bortzmeyer.org:33196;tag=3363661c
To: sip:beatrice@sip.durand.fr
Call-ID: 862152220@foobar.bortzmeyer.org
CSeq: 1 OPTIONS
Contact: sip:sipsak@foobar.bortzmeyer.org:33196
Content-Length: 0
Max-Forwards: 70
User-Agent: sipsak 0.8.11
Accept: text/plain

message received:
SIP/2.0 200 OK -- happy sipping to you asterisk
Via: SIP/2.0/UDP foobar.bortzmeyer.org:33196;rport=33196;received=213.41.181.9
From: sip:sipsak@foobar.bortzmeyer.org:33196;tag=3363661c
To: sip:beatrice@sip.durand.fr;tag=770257e1542af587787e4362acc86da0.7432
Call-ID: 862152220@foobar.bortzmeyer.org
CSeq: 1 OPTIONS
Content-Length: 0

** reply received after 82.232 ms **
SIP/2.0 200 OK -- happy sipping to you asterisk
final received

```

sipsak a utilisé OPTIONS, qui sert surtout au débogage ou bien à découvrir les capacités de l'“User Agent” d'en face, mais il y a aussi INVITE, BYE, etc. La réponse favorable identifiée par le nombre 200 ne surprendra pas les connaisseurs d'HTTP.

Vous avez peut-être également noté que les adresses SIP sont des URI. Ces URI sont décrites dans la section 19 de notre RFC (c'est un autre RFC, le RFC 3263, qui décrit en détail comment trouver un “User Agent” SIP à partir de son nom, par exemple en utilisant les enregistrements DNS SRV). Finis les numéros de téléphone peu pratiques à mémoriser, demain, pour la téléphonie comme pour le reste des applications Internet, on utilisera un nom de domaine et on pourra dire « Appelle-moi au sip:alain@voip.wengo.fr. ».

Fréquemment, les “User Agent” ne connaissent pas leur nom : si je veux appeler quelqu'un, je ne sais pas derrière quel ordinateur il est. SIP permet donc d'utiliser un mandataire (“proxy”). Ceux-ci sont typiquement gérés par des fournisseurs de service et fournissent d'autres services, par exemple de passerelle avec le réseau téléphonique classique. Ainsi, Wengo <<http://www.wengo.fr/>> est un tel fournisseur.

Le mandataire doit bien, lui, savoir quel est le nom actuel de l'“User Agent” du destinataire. Pour que cela soit possible, celui qui veut recevoir des appels doit joindre un “registrar” (en pratique, c'est typiquement le même service que le mandataire) et utiliser la commande SIP REGISTER pour indiquer sa localisation actuelle.

Comme le flux de données RTP doit pouvoir attendre la machine appelante, les NAT sont une des principales sources de problème pour SIP. La plupart des applications SIP utilisent STUN (RFC 3489) pour percer un trou dans le routeur NAT et permettre l'entrée des données.

Parmi les textes parlant de SIP en français, on peut consulter par exemple un projet d'étudiants de l'Université d'Avignon <<http://www.crir.univ-avignon.fr/pfe/sip0607/>> avec beaucoup de documentations.

SIP est aujourd'hui mis en œuvre dans plusieurs logiciels libres. Le programme SIP Communicator peut ainsi lancer et recevoir des appels via SIP. Le programme Asterisk est un auto-commutateur téléphonique tournant sur Unix et capable de faire du SIP. De nombreux autres logiciels SIP existent comme Twinkle ou Linphone.

Pour l'instant, SIP semble moins utilisé que ses concurrents fermés (et moins que la téléphonie classique). Mais cela peut changer dans le futur, au fur et à mesure que les utilisateurs comprennent les dangers qu'il y a à dépendre d'un seul fournisseur, utilisant un protocole non-standard et des logiciels dont on ne sait pas ce qu'ils font avec vos échanges téléphoniques. Là, SIP sera bien placé pour les remplacer.