RFC 6314 : NAT Traversal Practices for Client-Server SIP

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 23 juillet 2011

Date de publication du RFC : Juillet 2011

https://www.bortzmeyer.org/6314.html

Aujourd'hui, il est rare pour une machine individuelle (ordinateur de bureau, ordinateur portable, "*smartphone*", tablette, etc) d'avoir un vrai accès Internet. Presque toujours, la malheureuse machine est limitée à une adresse IP privée et coincée derrière un NAT qui traduira cette adresse à la volée. Cela ne gêne pas trop certains protocoles (SSH sortant, par exemple) mais est beaucoup plus gênant pour d'autres, notamment le protocole de téléphonie sur IP SIP. Ce RFC décrit l'ensemble des techniques que doivent mettre en œuvre les clients SIP pour arriver à passer quand même. C'est la meilleure lecture pour commencer à se pencher sur le problème « pourquoi a-t-on si souvent des problèmes avec SIP ? ».

Pour comprendre pourquoi SIP (RFC 3261 [1]) est particulièrement affecté, il faut se rappeler qu'il utilise des **références** : le client SIP indique au serveur à quelle adresse répondre, par exemple à quelle adresse envoyer les paquets audio du protocole compagnon RTP (RFC 3550). En présence d'un NAT, la machine ne connait pas son adresse IP telle que vue de l'extérieur et, de toute façon, le NAT traditionnel (qui agit sur le port, pas seulement sur l'adresse en dépit de son nom) bloque les paquets entrants, ne sachant pas que le flux RTP est lié à la connexion SIP. Pire, les différents routeurs NAT ont des comportements différents et il est donc difficile de trouver une solution qui marche partout (RFC 4787).

Ces problèmes du NAT sont bien connus et documentés. L'essentiel du code réseau d'un client SIP est consacré à contourner le NAT. Des solutions partielles ont été développées comme STUN (RFC 5389), TURN (RFC 5766), ICE (RFC 8445) ou bien des solutions spécifiques à SIP comme le RTP symétrique (RFC 4961), ou le SIP sortant (RFC 5626). Toutes ces solutions sont partielles et il manquait un document de haut niveau, décrivant le mécanisme général. Ce RFC 6314 ne propose donc pas « encore un nouveau protocole » mais explique la marche à suivre et les bonnes pratiques pour passer outre les routeurs NAT.

SIP peut être utilisé dans des contextes très différents : ce RFC se focalise sur l'utilisation en « client-serveur » où un client SIP (par exemple un "*softphone*") appelle un fournisseur SIP qui routera l'appel

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc3261.txt

vers le destinataire. Le SIP pair-à-pair (actuellement expérimental) n'utilisera pas forcément les mêmes techniques.

La section 3 détaille le problème que rencontrent les clients SIP en présence du NAT. Si le routeur NAT contient un ALG, celui-ci peut interférer avec SIP, par exemple en interdisant les communications chiffrées, ou bien en imposant que le même chemin soit emprunté à l'aller et au retour. Comme documenté dans les RFC 4787, RFC 3424 et RFC 8445, les ALG sont donc une solution limitée.

Et s'il n'y a pas d'ALG ? Dans ce cas, voyons comment SIP établit une connexion avec son fournisseur, par défaut. Le client envoie une requête sur UDP et le serveur est censé répondre au couple {adresse, port} de l'en-tête `Via:` de cet en-tête. Mais, après le passage du routeur NAT, cette adresse et son port, mis là par le client original, ne correspondent plus à rien d'utile. Même si le paquet de réponse atteint le NAT, il sera typiquement jeté par lui. Même si le client SIP s'était connecté en utilisant TCP, les appels qu'il recevra (`INVITE`) ne seront pas routés sur cette connexion TCP mais envoyés à l'adresse indiqué ce qui, là encore, ne marchera pas (sans compter d'autres problèmes comme la fermeture automatique de la session par le NAT après un délai de garde).

Cela, c'était pour la signalisation, pour le protocole SIP lui-même. Mais un coup de téléphone complet nécessite aussi de faire passer les données, les paquets audio. Ceux-ci voyagent en général via la protocole RTP (RFC 3550, le protocole à utiliser, les ports où envoyer les données, etc, étant indiqués via SDP (RFC 4566) ou bien via une requête SIP spécifique (RFC 3264). Même si la session SIP a été bien établie, ces paquets RTP risquent fort de ne pas arriver, les adresses et les numéros de port qu'indiquent SDP ou SIP n'ayant pas de signification en dehors du réseau local.

Donc, pour résumer cette section 3, SIP a deux problèmes avec le NAT :

— Pour la signalisation (SIP à proprement parler), requêtes entrantes et sortantes (pours des raisons différentes),
— Et pour les données (en général SDP + RTP).

Soyons maintenant positifs et voyons les solutions (section 4). Pour la signalisation, la section 4.1 expose les méthodes possibles :

— Pour les requêtes sortantes, la réponse symétrique, où la réponse est envoyée au port d'origine (comme le font d'autres protocoles UDP comme le DNS), port ouvert sur le NAT par l'envoi de la requête. Cette méthode est décrite dans le RFC 3581.
— Pour les requêtes entrantes (RFC 5626), le fournisseur SIP doit garder trace du port utilisé et s'en servir pour acheminer les appels entrants.

Pour le transport des données, la section 4.2 cite :

— Le RTP symétrique (RFC 4961) qui permet d'ouvrir un trou dans le NAT depuis l'extérieur, avant de demander au pair d'y envoyer le trafic.
— RTCP (section 6 du RFC 3550) et le RFC 3605 pour indiquer le port à utiliser.
— Et surtout la solution totale, la combinaison de STUN (RFC 5389) et TURN (RFC 5766), orchestrés par ICE (RFC 8445). STUN permet de découvrir l'adresse IP publique qui va être utilisée pour les communications externes, TURN va relayer le trafic pour les cas où le routeur NAT est particulièrement désagréable et empêche STUN de fonctionner (TURN est très coûteux en ressources puisque **chaque** paquet de données doit être ainsi relayé, mais il marche dans tous les cas), et ICE a pour rôle d'essayer successivement et intelligement plusieurs méthodes de traversée de NAT (STUN, TURN et d'autres) pour choisir la plus adaptée.

La section 5 illustre ensuite ces méthodes par divers exemples. En voici un très simple, pour la signalisation, où le client, `192.168.1.2` est derrière un NAT (qui réécrit son adresse en `172.16.3.4`) et s'enregistre auprès d'un fournisseur SIP, Example.com. La requête est :

```
REGISTER sip:example.com SIP/2.0
Via: SIP/2.0/UDP 192.168.1.2;rport;branch=z9hG4bKnashds7
Max-Forwards: 70
From: Bob <sip:bob@example.com>;tag=7F94778B653B
To: Bob <sip:bob@example.com>
Call-ID: 16CB75F21C70
CSeq: 1 REGISTER
Supported: path, outbound
Contact: <sip:bob@192.168.1.2 >;reg-id=1
    ;+sip.instance="<urn:uuid:00000000-0000-1000-8000-AABBCCDDEEFF>"
Content-Length: 0
```

Le client SIP a utilisé le paramètre `rport` (RFC 3581) dans l'en-tête `Via:` pour indiquer que le serveur doit répondre au port d'où vient la requête (et que le routeur NAT a ouvert). La réponse est :

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.1.2;rport=8050;branch=z9hG4bKnashds7;
     received=172.16.3.4
From: Bob <sip:bob@example.com>;tag=7F94778B653B
To: Bob <sip:bob@example.com>;tag=6AF99445E44A
Call-ID: 16CB75F21C70
CSeq: 1 REGISTER
Supported: path, outbound
Require: outbound
Contact: <sip:bob@192.168.1.2 >;reg-id=1;expires=3600
     ;+sip.instance="<urn:uuid:00000000-0000-1000-8000-AABBCCDDEEFF>"
Content-Length: 0
```

et le paramètre `received` du champ `Via:` indique l'adresse IP source qu'avait vue le serveur. (Une autre solution aurait été d'utiliser TCP.) Les amateurs de scénarios plus complexes seront comblés, avec le reste de la longue section 5.

À noter qu'un argument souvent présenté en faveur de Skype, par rapport à SIP, est son meilleur comportement en présence de NAT hostiles (parfois très hostiles, par exemple ne laissant passer que HTTP). Il est amusant de noter que deux des auteurs du RFC travaillent chez Skype. Mais, surtout, si les clients SIP ont souvent moins de succès que Skype lors de la traversée de routeurs très fermés, c'est pour trois raisons :

— Tous les logiciels SIP ne mettent pas encore en œuvre l'intégralité des recommandations résumées dans ce RFC 6314. La qualité des clients et serveurs SIP existants est très variable.
— Skype utilise également des méthodes non-standard (par exemple il peut tunneler la voix sur HTTP, bricolage très contestable mais qui est parfois la seule voie restée ouverte),
— Et Skype utilise également des méthodes de parasite, comme d'enrôler comme relais des machines qui n'ont rien demandé <`http://www.velocityreviews.com/forums/t727931-skype-fixed-sup` `html`> ce qui, à juste titre, suscite des protestations <`http://www.its.monash.edu.au/staff/` `networks/about/policies/skype.html`>.