

RFC 5389 : Session Traversal Utilities for (NAT) (STUN)

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 25 Octobre 2008

Date de publication du RFC : Octobre 2008

<http://www.bortzmeyer.org/5389.html>

Le NAT a toujours été une des plaies de l'Internet, entre autres parce qu'il perturbe les applications qui veulent transmettre une référence à leur adresse IP. STUN, décrit dans ce RFC, est un des protocoles qui permet de limiter les dégâts.

Pour plusieurs raisons, dont le manque d'adresses IPv4, de nombreuses machines sont connectées à l'Internet derrière un routeur qui fait du NAT. Leur adresse étant alors privée, elles ne peuvent pas la transmettre à l'extérieur, ce qui est une gêne considérable pour tous les protocoles qui reposent sur la référence à des adresses IP, comme SIP. SIP lui-même (RFC 3261¹) marche à travers le NAT mais les données audio ou vidéo transmises (typiquement par RTP) le sont à une adresse IP que l'appelant donne à l'appelé et, si cette adresse est privée, le flux multi-média n'arrivera jamais.

La bonne solution serait de déployer IPv6, qui fournit des adresses en quantité illimitée, ou à la rigueur de permettre un meilleur contrôle du routeur NAT avec MIDCOM (RFC 3303) mais, en attendant, STUN est une solution simple et qui marche souvent. STUN avait à l'origine été normalisé dans le RFC 3489, que ce RFC met à jour, avec des changements assez importants, mais qui n'empêchent pas la compatibilité des anciens clients avec les nouveaux serveurs (ou réciproquement, sauf si le client utilise les nouvelles possibilités, comme la possibilité de fonctionner sur TCP et plus seulement UDP).

STUN s'inscrit donc dans la catégorie des protocoles de « réparation » du NAT, catégorie décrite dans le RFC 3424.

STUN résoud partiellement le problème des NAT de la manière suivante : un serveur STUN doit être installé quelque part sur l'Internet public (il existe des serveurs STUN publics <<http://www.vovida.org/applications/downloads/stun/>>) et reçoit des demandes envoyées en UDP (ou TCP, voir la

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

description de cette nouveauté en section 7.2.2) au port 3478. Le serveur STUN peut alors connaître l'adresse publique, celle mise par le routeur NAT et, en répondant au client, il pourra informer celui-ci « Voilà à quoi tes paquets ressemblent, vus de l'extérieur ».

Le nom du serveur STUN est typiquement configuré dans le client, puis le serveur recherché dans le DNS via les enregistrements SRV (RFC 2782), comme détaillé dans la section 9. (Le client Unix en ligne de commande ne connaît pas encore les SRV.)

Le principe est simple, mais le RFC est plus compliqué que cela, notamment en raison des problèmes de sécurité (la section 16 du RFC est très détaillée sur ce point). Par exemple, le client STUN doit en fait commencer par une connexion TCP pour obtenir un mot de passe temporaire. (d'autres méthodes d'authentification sont possibles.)

Un autre problème est qu'il y a en fait plusieurs sortes de NAT (voir par exemple le RFC 2663). Par exemple, certains routeurs NAT ("*Symmetric NAT*" pour le RFC 3489) n'attribuent pas l'adresse externe uniquement en fonction de la **source** mais aussi en fonction de la **destination**. Ceux-ci ne fonctionnent pas avec STUN, qui a besoin de NAT "*cone*", c'est-à-dire où les paquets d'une même machine auront toujours la même adresse IP externe.

Voici une démonstration d'un client STUN <<http://sourceforge.net/projects/stun/>> qui, en se connectant à un serveur STUN public <<http://www.voip-info.org/wiki-STUN>> va apprendre son adresse IP publique :

```
% ifconfig -a
hme0: [...] inet 172.19.1.2
(Adresse privée, probablement NATée...)

% ./client stun.ekiga.net -v |& more
STUN client version 0.96
...
MappedAddress = 213.41.181.9:32143
(Voici mon adresse telle que vue de l'extérieur, à noter que ce RFC,
contrairement à son prédécesseur la nomme désormais ReflexiveAddress)
...
```

Une fois l'adresse extérieure détectée, tout n'est pas terminé car rien n'indique, par exemple, que les paquets vont effectivement passer. C'est pour cela que la section 14 insiste que STUN n'est qu'un outil, devant s'insérer dans une solution plus globale, comme ICE (pas encore normalisé). À lui seul, STUN ne permet pas la traversée de tous les NAT. La section 14 décrit en détail ce concept d'utilisation de STUN et les règles que doivent suivre les protocoles qui utilisent STUN.

La section 6 décrit le format des paquets. Un paquet STUN doit comprendre l'indication d'une **méthode**, qui indique le genre de services que désire le client. Notre RFC 5389 ne décrit qu'une seule méthode, *Binding*, qui permet d'obtenir son adresse IP extérieure mais TURN (RFC 5766), par exemple, en définit d'autres. Après la méthode et un identificateur de transaction (qui sert au serveur STUN à séparer ses clients), suivent les **attributs**, encodés en TLV (la liste des attributs figure en section 15). Les numéros des attributs sont compris entre 0x0000 et 0x7FFF s'ils doivent être reconnus par le serveur (autrement, la requête est rejetée) et entre 0x8000 et 0xFFFF si leur compréhension est facultative (cette notion d'attributs obligatoires ou facultatifs facilite les évolutions ultérieures du protocole). Une partie de l'identificateur de transaction de la version précédente de STUN a été transformé en "*magic cookie*", une valeur fixe (0x2112A442) qui sert à reconnaître les agents STUN conformes à la nouvelle version.

Les sections 2, 12 et 19 décrivent en détail les changements par rapport à la norme précédente, le RFC 3489, notamment le fait que STUN (qui a changé de nom au passage, en gardant le même sigle) n'est plus présenté comme une solution complète mais comme un outil pour une solution plus générale (qui doit, notamment, tester que la connectivité fonctionne, comme le fait ICE). Même la découverte du comportement du NAT est désormais effectuée par des protocoles plus riches comme celui du RFC 5780. Parmi les autres changements, il faut noter que STUN encode désormais les adresses IP (par un simple XOR, section 15.2) pour les masquer, empêchant ainsi des stupides routeurs NAT de les réécrire (oui, certains routeurs NAT, profitant de l'absence de mécanisme de responsabilité pour le logiciel embarqué, examinaient toutes les données transitant par eux pour réécrire les adresses IP afin d'y mettre l'adresse publique ; le RFC parle poliment de "*misguided attempt at providing a generic ALG function*" mais le terme correct est « débilite monstrueuse »).