RFC 6156 : Traversal Using Relays around NAT (TURN) Extension fo IPv6

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 28 avril 2011

Date de publication du RFC : Avril 2010

https://www.bortzmeyer.org/6156.html

Le protocole TURN, qui permet à deux machines coincées derrière des routeurs NAT de communiquer en utilisant un relais tiers, a été normalisé dans le RFC 5766[1], dans une version minimale, uniquement pour IPv4 et seulement pour UDP, les services « incontestables », qui étaient pressés d'avoir une solution. Depuis, des extensions à TURN sont développées pour d'autres services. C'est ainsi que ce RFC 6156 permet d'utiliser TURN pour établir des liaisons en IPv6. Cette extension à TURN a depuis été intégrée dans la norme de base, le RFC 8656, remplaçant ce RFC.

Cette extension permet de communiquer avec une machine IPv4 depuis une machine IPv6, avec une machine IPv6 depuis une autre machine IPv6 et enfin avec une machine IPv6 depuis une machine IPv4 (section 1 du RFC). C'est donc une véritable solution de communication entre les deux protocoles, travaillant dans la couche 7 (contrairement aux techniques comme le NAT qui travaillent dans la couche 3). Le serveur TURN relais la totalité des données entre les deux clients TURN.

On peut se demander si un tel service est bien nécessaire. Après tout, TURN avait été inventé pour le cas du NAT, qui lui-même était motivé par le faible nombre d'adresses IPv4 disponibles. En IPv6, le problème de la pénurie ne se pose pas et TURN semble inutile. Toutefois, comme le note le RFC 5902, il risque d'y avoir quand même des réseaux IPv6 NATés, peut-être avec des adresses ULA à l'intérieur, et TURN peut donc trouver sa place ici. Autre utilité : TURN peut aider les machines purement v4 (la majorité d'aujourd'hui) et les machines purement v6 (qui se répandront sans doute) à communiquer entre elles. Enfin, TURN peut aussi être utile pour un administrateur de pare-feu à état pour permettre aux utilisateurs d'ouvrir des ports afin de recevoir des connexions entrantes. TURN a été conçu pour qu'il soit très restreint dans ce qu'il accepte, de façon à ce que les administrateurs de pare-feux puissent lui faire confiance.

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc5766.txt

La nouvelle extension fonctionne avec un nouvel attribut TURN, `REQUESTED-ADDRESS-FAMILY`, valeur 0x0017 (section 3). L'attribut `XOR-RELAYED-ADDRESS` (RFC 5766, section 14.5, et section 4.2 de notre RFC) de la réponse contiendra une adresse de la famille demandée (v4 ou v6). La famille d'adresse du client est déterminée, elle, en regardant l'adresse utilisée pour contacter le serveur. À noter que chaque demande d'allocation TURN ne peut gérer qu'une seule adresse IP. Le client qui voudrait communiquer avec une adresse IPv4 et une v6 devra faire deux demandes d'allocation. Si le serveur ne veut ou ne peut pas gérer une famille donnée, il répondra 440 (`Address Family not Supported`).

La section 4 contient ensuite les détails concrets comme le format du nouvel attribut (section 4.1.1). Celui-ci a le numéro 0x0017, ce qui le met dans les plages des attributs dont la compréhension par le serveur est impérative. Un vieux serveur qui ne met pas en œuvre notre RFC ne peut donc pas ignorer cet attribut.

Un problème qui a toujours été très chaud dans le groupe Behave <`https://www.bortzmeyer.org/behave-wg.html`> est celui de la traduction des paquets. Comme on dit en italien, « "*Traduttore, Traditore*" » (traducteur = traître). Lorsqu'un serveur TURN relaie des paquets, doit-il respecter toutes les options de l'en-tête IP ? La question est évidemment encore plus difficile pour une traduction entre IPv4 et IPv6. La section 8 fournit les règles. Par exemple, lors de la traduction de IPv4 vers IPv6 (section 8.1), que doit valoir le champ "*Traffic Class*" (RFC 2460, section 7) d'un paquet IPv6 sortant ? Le comportement souhaité est celui spécifié dans la section 3 du RFC 7915 (RFC qui décrit le mécanisme de traduction de IPv4 vers IPv6, notamment pour l'utilisation dans NAT64). Le comportement possible est d'utiliser la valeur par défaut de ce champ. Le "*Flow label*" (RFC 2460, section 6), lui, devrait être mis à zéro (il n'a pas d'équivalent IPv4). Et pour les en-têtes d'extensions IPv6 (RFC 2460, section 4) ? Aucun ne doit être utilisé, dans tous les cas (sauf la fragmentation).

Et si on relaie entre deux machines IPv6 ? Ce sont quasiment les mêmes règles, comme indiqué par la section 8.2. Par exemple, bien qu'il existe un "*Flow label*" des deux côtés, on considère que TURN, relais de la couche 7, ne devrait pas essayer de le copier : il y a deux flots différents de part et d'autre du serveur TURN.

Cette extension de TURN pose t-elle des problèmes de sécurité ? Potentiellement oui, répond la section 9. En effet, elle donne aux machines des capacités nouvelles. Ainsi, une machine purement IPv6 obtient, si elle a le droit d'accéder à un serveur TURN mettant en œuvre ce RFC 6156, la possibilité de parler aux machines purement IPv4, possibilité qu'elle n'avait pas avant. Comme il est recommandé de ne faire fonctionner TURN qu'avec une authentification préalable des clients, ce n'est sans doute pas une question sérieuse en pratique.

La section 10 décrit l'enregistrement à l'IANA des nouveaux paramètres (attribut `REQUESTED-ADDRESS-FAMILY` et codes d'erreur 440 et 443) dans le registre des paramètres STUN <`https://www.iana.org/assignments/stun-parameters/stun-parameters.xhtml`> (rappelez-vous que TURN est défini comme une extension de STUN).

Question mise en œuvre, il en existe une côté serveur, non distribuée mais accessible vie le réseau (voir <`http://numb.viagenie.ca/`>), une libre <`http://turnserver.sourceforge.net/`> qui a IPv6 depuis la version 0.5, mais apparemment rien encore côté client.

Merci à Simon Perreault pour sa relecture et ses remarques.