RFC 5245 : Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 30 avril 2010

Date de publication du RFC : Avril 2010

https://www.bortzmeyer.org/5245.html

Le problème de la traversée des routeurs NAT continue à susciter une grande activité normalisatrice, de façon à réparer la grossière erreur qu'avait été le déploiement massif du NAT plutôt que celui de IPv6. ICE est un « méta-protocole », orchestrant plusieurs protocoles comme STUN et TURN pour arriver à découvrir un canal de communication malgré le NAT. Spécifié à l'origine dans ce RFC, il est désormais normalisé dans le RFC 8445 [1].

L'objectif principal est la session multimédia, transmise entre deux terminaux, en général en UDP. ICE sera donc surtout utilisé au début par les solutions de téléphonie sur IP. Ces solutions ont déjà dû affronter le problème du NAT et ont développé un certain nombre de techniques, dont la bonne utilisation est l'objet principal de notre RFC.

En effet, les protocoles de téléphonie sur IP, comme SIP (RFC 3261) ont en commun d'avoir une session de **contrôle** de la connexion, établie par l'appelant en TCP (et qui passe en général assez bien à travers le NAT, si l'appelé utilise un relais public) et une session de transport des **données**, en général au dessus de UDP. C'est cette session qui est en général perturbée par le NAT. La session de contrôle transmet au partenaire SIP l'adresse IP de son correspondant mais, s'il y a deux domaines d'adressage séparés (par exemple un partenaire sur le domaine public et un autre dans le monde des adresses privées du RFC 1918), cette adresse IP ainsi communiquée ne sert pas à grand'chose.

On voit donc que le non-déploiement d'IPv6, qui aurait permis de se passer du NAT, a coûté très cher <https://www.bortzmeyer.org/ipv6-et-l-echec-du-marche.html> en temps de développement.

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc8445.txt

Notre RFC fait 119 pages, et se traduira par du code réseau très complexe, uniquement pour contourner une mauvaise décision.

Des solutions au NAT, soit standard comme STUN (RFC 5389) soit purement spécifiques à un logiciel fermé comme Skype ont été développées. Mais aucune n'est parfaite, car tous les cas sont spécifiques. Par exemple, si deux machines sont sur le même réseau local, derrière le même routeur NAT, faire appel à STUN est inutile, et peut même échouer (si le routeur NAT ne supporte pas le routage en épingle à cheveux). Le principe d'ICE est donc de décrire comment utiliser plusieurs protocoles de traversée de NAT, pour trouver la solution optimale pour envoyer les paquets de la session de données.

Le principe d'ICE, exposé dans la section 2 et détaillé dans la 4, est donc le suivant : chaque partenaire va déterminer une liste de paires d'adresses de transport candidates (en utilisant leurs adresses IP locales, STUN et TURN), les tester et se mettre d'accord sur la « meilleure paire ». Une adresse de transport est un couple (adresse IP, port).

La première étape est donc d'établir la liste des paires, et de la trier par ordre de priorité. La section 4.1.2.1 recommande une formule pour calculer la priorité, formule qui fait intervenir une préférence pour le type d'adresses IP (une adresse locale à la machine sera préférée à une adresse publique obtenue par STUN, et celle-ci sera préférée à l'adresse d'un serveur relais TURN) et une préférence locale, par exemple pour les adresses IPv6 par rapport à IPv4. On notera donc (section 4.1.2.2) qu'ICE facilitera le fonctionnement des machines à double pile (v4 et v6) en fournissant un moyen simple de préférer une des deux familles, tout en pouvant se rabattre sur l'autre.

La deuxième étape est de tester les paires (section 2.2 et 7) pour déterminer celles qui marchent. ICE permet d'arrêter les tests dès le premier succès ("*aggressive nomination*") ou bien de les poursuivre, pour voir si un meilleur RTT peut être trouvé ("*regular nomination*").

La troisième et dernière étape d'ICE est de sélectionner la meilleure paire (en terme de priorité) parmi celles qui ont fonctionné. Les couples (adresse IP, port) de celles-ci seront alors utilisées pour la communication.

L'ensemble du processus est relativement compliqué et nécessite de garder un état sur chaque partenaire ICE, alors qu'ICE est inutile pour des machines qui ont une adresse IP publique. La section 2.7 introduit donc la possibilité de mises en œuvres légères ("*lite implementation*") d'ICE, qui peuvent interagir avec les autres machines ICE, sans avoir à gérer tout le protocole.

Tous ces tests prennent évidemment du temps, d'autant plus de temps qu'il y a de paires d'adresse de transport « nominées ». C'est le prix à payer pour la plus grande souplesse d'ICE : il sera toujours plus lent que STUN seul. La section 12.1 se penche donc sur ce problème et suggère de ne pas attendre le dernier moment pour commencer les tests ICE. Par exemple, un téléphone matériel peut les commencer dès qu'il est décroché, sans attendre la composition du numéro.

Les protocoles de téléphonie sur IP ayant eu leur part de vulnérabilités, la section 18, sur la sécurité, est très détaillée. Par exemple, une attaque classique est d'établir une communication avec un partenaire, puis de lui demander d'envoyer le flux de données vers la victime. C'est une attaque par amplification classique, sauf que l'existence d'une session de données séparée de la session de contrôle fait qu'elle ne nécessite même pas de tricher sur son adresse IP (et les techniques du RFC 2827 sont donc inefficaces). Cette attaque, dite « attaque du marteau vocal », peut être combattue grâce à ICE, puisque le test de connectivité échouera, la victime ne répondant pas (puisqu'elle n'a rien demandé). Si tout le monde utilise ICE, cette attaque peut donc complètement disparaitre.

D'innombrables détails compliquent les choses et expliquent en partie la taille de ce RFC. Par exemple, la section 10 décrit l'obligation d'utiliser des "*keepalives*", des paquets inutiles mais qui ont pour seul but de rappeler au routeur NAT que la session sert toujours, afin que les correspondances entre une adresse IP interne et une adresse externe restent ouvertes (le flux de données ne suffit pas forcément, car il peut y avoir des périodes d'inactivité). Tous les protocoles de la session de données ne permettant pas forcément d'envoyer des paquets « bidon », notre RFC suggère même d'envoyer délibèrement des paquets incorrects, pour faciliter leur élimination par le partenaire.

Enfin, une intéressante section 20 décrit les problèmes pratiques du déploiement (c'est une préoccupation rare dans les RFC). Par exemple, la planification de la capacité des serveurs est discutée en 20.2.1. Un serveur STUN n'a pas besoin de beaucoup de bande passante, mais un serveur TURN, oui, puisqu'il relaie tous les paquets, y compris ceux de la session de données.

La première version d'ICE ne gérait que l'UDP mais, depuis la publication du RFC 6544, TCP est également accepté.

Il existe déjà au moins une implémentation, pjnath <http://blog.pjsip.org/2007/04/06/introducing-pjnath-open-source-ice-stun-and-turn/>.