Influence des timers TCP de dispositifs NAT sur l'expérience utilisateur

Isabelle Kraemer Frédéric Perrin

7 janvier 2011

1 Introduction

Avec l'épuisement des plages d'adresses IPv4 et un déploiement d'IPv6 qui reste très limité, les opérateurs d'accès à Internet doivent rechercher des solutions permettant d'offrir à une clientèle toujours plus nombreuse un accès à l'Internet IPv4, tout en limitant la consommation d'adresses IPv4. Cet ensemble de contraintes a d'abord obligé les opérateurs à attribuer des adresses dynamiques, une par foyer. Cependant, cela n'est pas suffisant ; il faut partager la même adresse IPv4 entre plusieurs foyers.

L'une des technologies permettant cela est l'architecture DS-Lite (pour *Dual-Stack Lite*), permettant le partage d'une unique adresse IPv4. Le nombre de connexion simultanées traversant un NAT est limité par l'étendue de la plage de port attribuée aux clients. Cet article explore les conséquences, du point de vue de l'utilisateur, d'un sous-dimensionnement de cette plage de ports.

2 Terminologie

DS-Lite Pour *Dual-Stack Lite*. Une architecture de réseau d'opérateur permettant de gérer le manque d'adresses IPv4 publiques. Le réseau de collecte et de cœur de l'opérateur fonctionne uniquement en IPv6[2].

B4 Pour *Basic Bridging BroadBand element*. Modem installé chez l'utilisateur final. Il distribue des adresses IPv4 privées aux équipements du client, ainsi que des adresses IPv6. Pour offrir un accès vers l'Internet IPv4, le B4 établit un tunnel IPv4-dans-IPv6 jusqu'à l'AFTR de l'opérateur.

AFTR Pour *Address Family Transition Router*. CGN installé par l'opérateur. Son fonctionnement est proche de celui d'un NAT classique, si ce n'est la conservation d'un élément supplémentaire dans le contexte du NAT : l'adresse IPv6 du tunnel d'origine de la connexion, qui identifie le B4 (et donc le client) initiateur de la connexion.

CGN Pour *carrier-grade NAT*. Un NAT géré par l'opérateur, et partagé par un nombre potentiellement élevé d'utilisateurs finaux.

3 Travaux précédents

Ce projet est dans la continuité du projet de 3eannée de Florent Fourcot et Bertrand Grelot, clos en mars 2010, dont le but était l'étude du comportement d'un NAT à grande échelle, particulièrement en ce qui concerne la consommation de ports[3].

4 Environnement de test

Nous disposons d'un élément B4 de marque Netgear, installé dans la résidence des étudiants (modèle WNDR3700, . L'AFTR est une machine Fedora GNU/Linux 13 (noyau 2.6.34), installée dans le laboratoire RSM de l'École. Le dæmon AFTR utilisé est celui de l'ISC, développé par Francis Dupont[1]. Le réseau entre le B4 et l'AFTR route nativement l'IPv6 ; le chemin entre le B4 et l'AFTR est successivement sous les autorité du Réseau des Élèves, de la DISI de l'École et du laboratoire RSM de l'École.

Seules quelques machines, celles utilisées par les auteurs, sont placées derrière le B4. Afin de se placer dans une situation de sous-dimensionnement de la plage de ports attribuée aux clients, nous réduisons artificiellement cette plage.

5 Tests menés

Florent Fourcot et Bertrand Grelot [3] ont mesuré la consommation de ports lors de la navigation sur des sites « Web 2.0 ». Ainsi, la navigation sur Facebook consomme environ 50 ports par page ouverte. Nous configurons donc l'AFTR pour utiliser une plage de ports de 100. Ensuite, un client derrière le B4 commence une navigation sur Facebook (machine Ubuntu GNU/Linux 10.04, Firefox 3.6.13). Les paramètres suivants ont été configurés :

- `network.http.pipelining` à `false` (valeur par défaut) ;
- `network.http.use-cache` à `false`, et nettoyage du cache ;
- `network.dns.cacheExpiration` à 0 ;
- `network.dns.disableIPv6` à `true`, pour forcer le trafic à passer par le tunnel du B4.

Dès la deuxième page, on remarque que certains éléments graphiques de la page sont très longs à apparaître (de l'ordre de deux minutes). Du point de vue de l'utilisateur, le seul effet visible du manque de ports est l'absence de réponse.

Lorsque la plage de ports est épuisée, différents programmes réagissent différemment. Comme nous venons de le voir, Firefox ne va afficher à l'utilisateur que des éléments manquants (voire une page blanche, si même la requête de la page n'a pas trouvé de port disponible), jusqu'à ce qu'un port se libère. D'autres logiciels, comme le client OpenSSH, abandonneront la connexion après plusieurs dizaines de secondes, avec le message standard « *connection timed out* ».

Une observation des trames réseau montre que l'AFTR ne traduit pas et ne fait pas suivre les segments TCP `SYN` envoyés par les clients après l'épuisement

des ports. Aucun paquet ICMP d'erreur n'est renvoyé aux clients. Ne recevant pas de réponse ni d'erreur, les clients retransmettent les segments qu'ils considèrent comme perdus, jusqu'à libération d'un port sur l'AFTR ou déclenchement d'un minuteur.

6 Résultat des tests

Intuitivement, on pourrait penser que le problème d'insuffisance de port est marginal. En effet, en comptant 50 images dans une page et 50 ms pour télécharger une image, un seul port serait suffisant pour afficher une page : en établissant successivement 50 connexions TCP (une pour chaque image), il serait à nouveau disponible après 2,5 s seulement. Mais la réalité est toute autre, comme nous avons pu le constater dans nos tests.

6.1 TCP

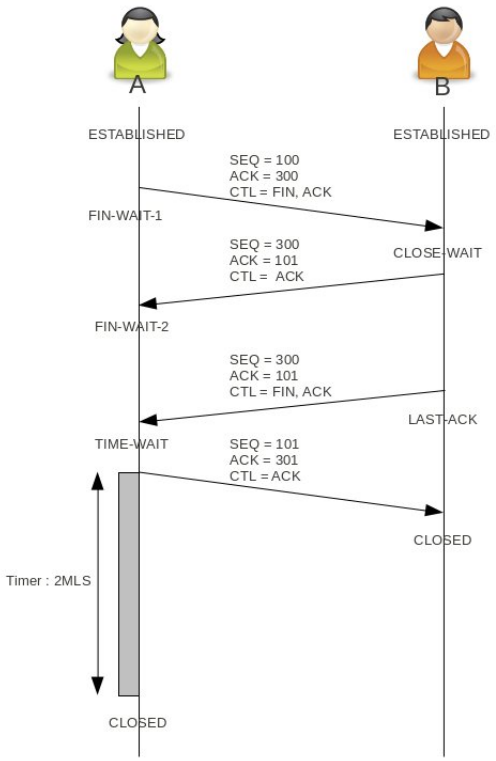
Pour interpréter les tests menés, il faut d'abord bien comprendre le protocole utilisé, TCP[4], et plus particulièrement, il faut comprendre les mécanismes en jeu lors de la fermeture d'une connexion TCP. Considérons le cas où une connexion est établie entre A et B. Supposons que A veuille terminer la communication.

1. Dans un premier temps, A envoie un paquet FIN.
2. La deuxième partie, B, acquitte la demande de fermeture de connexion avec un ACK. La connexion est à présent fermée dans le sens A vers B. À partir de maintenant, A ne peut plus envoyer de données. Par contre, il est possible que B continue à émettre des données.
3. B peut continuer à envoyer des données à A. Puis, lorsque B souhaite également terminer la connexion, il envoie à son tour un FIN.
4. Finalement, A envoie un ACK qui clôture la connexion.

Les acquittements des paquets de fermeture des connexions ne sont pas eux-mêmes acquittés. Imaginons que lors de la quatrième étape, le ACK

Ainsi, dans l'éventualité où l'acquittement d'un FIN est perdu, A doit laisser la connexion ouverte suffisamment longtemps pour recevoir une éventuelle retransmission du FIN. Cela explique qu'un timer TCP pour l'état TIME-WAIT ait été mis en place : à la fin du timeout, la connexion est fermée de toute manière. En conséquence, un port ne peut pas être réutilisé de suite, alors que la connexion vient de se fermer. Le timeout associé à l'état TIME-WAIT dure 2 MSL, ce qui correspond à 60 secondes sous Linux.

Un récapitulatif est présenté dans le schéma ci-dessous. Figurent les échanges de paquet et l'évolution de l'état de la connexion pour A et pour B sur leurs lignes de vie respectives.



Différentes formes de fermeture des connexions TCP existent, mais nécessitent toujours le blocage final sur une minuterie.

Pourtant, lors de nos tests, nous avons constaté, grâce à la commande `nstat`, que des ports sortaient de l'état TIME-WAIT mais n'était réutilisés qu'après une latence. La fermeture de connexion de TCP ne suffit donc pas à expliquer ce que nous avons pu observer.

6.2 NAT

En effet, il ne faut pas oublier que les connexions s'effectuent derrière un NAT. Ce-dernier établit un contexte lors de chaque connexion, c'est-à-dire que, par précaution, il ajoute un timer supplémentaire lors de la fin d'une communication. Dans les paramètres du programme, on trouve en conséquence les indications suivantes : `#define CLOSED_TCP_LIFETIME 120` et `#define UDP_LIFETIME 300`.

Lors de la fermeture d'une connexion TCP, un port reste réservé pendant 2 minutes, et lors d'une communication UDP, le port est inattribué pendant 5 minutes à compter du dernier paquet émis/reçu.

6.3 Interprétation

Maintenant, nous avons toutes les cartes en main pour expliquer nos observations. Lorsque le nombre de ports est suffisant, un port est attribué dès qu'une connexion est demandée, la latence est celle du réseau : les éléments s'affichent sans problème, l'expérience utilisateur est bonne. Par contre, lorsqu'il n'y a pas assez de port, le NAT ne peut donner suite aux demandes de connexion du client : les paquets sont perdus. Sans réponse, le client continue d'émettre des paquets. Finalement, un port finit par se libérer, ce qui permet d'établir la connexion. L'expérience utilisateur est très mauvaise : affichage très lent des éléments graphiques, connexion non fluide... La difficulté est maintenant de mesurer la vitesse de dégradation de l'expérience utilisateur. **Comment mesure-t-on l'expérience utilisateur, quels sont les paramètres pertinents ? Comment simule-t-on plusieurs utilisateurs qui utilisent leur connexion Internet en même temps ?**

Références

[1] F. Dupont : The AFTR Daemon. `http://www.isc.org/software/aftr`.

[2] A. Durand, R. Droms, J. Woodyatt et Y. Lee : Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion, août 2010. Internet-Draft. `http://tools.ietf.org/html/draft-ietf-softwire-dual-stack-lite-06`.

[3] Bretrand Grelot et Florent Fourcot : *Migrating to IPv6 with Address+Port translation*. Rapport de projet SLR, Télécom Bretagne, mars 2010.

[4] J. Postel : RFC 793 : Transmission Control Protocol, septembre 1981. `http://tools.ietf.org/html/rfc793`.