RFC 6708 : Application-Layer Traffic Optimization (ALTO) Requirements

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 30 septembre 2012

Date de publication du RFC : Septembre 2012

https://www.bortzmeyer.org/6708.html

Le système Alto ("*Application-Layer Traffic Optimization*"), permettant aux machines d'un réseau pair-à-pair de trouver le meilleur pair, par exemple pour obtenir un fichier plus rapidement, avance, lentement, mais sûrement. Ce nouveau RFC spécifie le cahier des charges d'Alto.

Alto peut servir à autre chose qu'au pair-à-pair. Mais le scénario d'utilisation principal est le suivant : une machine veut accéder à un fichier. Celui-ci est offert par plusieurs pairs. Lequel choisir ? En prendre un au hasard risque de ne pas être optimum, ni pour la machine (temps de transfert du fichier plus long qu'il ne pourrait l'être), ni pour son FAI (utilisation de liens externes plus coûteux que son réseau interne). Alto vise donc à permettre à des serveurs Alto de donner à leurs clients (les machines qui veulent accéder aux fichiers) des informations sur lesquelles décider intelligemment, ou même simplement d'indiquer directement le pair à utiliser. Le groupe de travail Alto <https://www.bortzmeyer.org/alto-wg.html> a été créé en novembre 2008, le RFC 5693 [1], qui définit le problème à résoudre, a été publié en octobre 2009, et voici désormais le deuxième RFC du groupe, le cahier des charges du protocole. (Pour ceux qui ne connaissent pas l'IETF, et qui craignent que le travail sur le protocole ne commence qu'une fois le cahier des charges terminé, je vous rassure, ce n'est pas ainsi que fonctionne l'IETF. Le protocole est déjà bien avancé à l'époque, et a été publié dans le RFC 7285.)

Pair-à-pair ou pas, le but d'Alto est donc d'améliorer simultanément la qualité de la vie pour l'utilisateur (télécharger les chefs d'œuvre de la culture plus vite) et pour le réseau (ne pas gaspiller des ressources chères comme les lignes transocéaniques). Cela ne peut pas se faire par des mesures uniquement faites par les machines terminales. Par exemple, celles-ci peuvent faire des ping mais elles ne

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc5693.txt

peuvent pas déterminer si un lien est de "*peering*" (« gratuit ») ou de transit (facturé au 95ème centile de l'usage).

Les serveurs Alto ne feront pas bouger les octets : leur seul rôle est de dire aux pairs « tu devrais plutôt causer avec lui, ça devrait aller vite ». Les pairs sont ensuite responsables de leurs décisions.

L'habituelle section de terminologie, nécessaire pour comprendre le reste du RFC, est la section 2. Attention, le RFC 5693 définissait déjà certains termes (comme client Alto et serveur Alto) qui ne sont pas répétés ici. Les termes les plus importants :

— Descripteur de groupe ("*Host Group Descriptor*") : c'est l'information qui définit un groupe de machines. Cela peut être un préfixe IP (« toutes les machines en `2001:db8:1337::/32` ») ou un numéro d'AS (« toutes les machines joignables par l'AS 42 »).
— Attribut d'une machine ("*Host Characteristics Attribute*") : c'est une propriété attachée à une machine comme « accès Internet au forfait ».
— Critère de classement. On a dit plus haut que le but d'Alto était de sélectionner le « meilleur » pair. Mais quelle est la définition de « meilleur » ? Par exemple, cela peut être « le moins cher » ou « celui avec la plus forte capacité réseau ». Alto est un protocole pour distribuer des choix, pas un mécanisme pour décider quel est le meilleur.
— Recherche dépendante de la cible ("*Target-aware Query Mode*") se dit d'une requête Alto où le client Alto connait une liste de pairs potentiels, envoie cette liste au serveur Alto (ainsi qu'un critère de classement) et reçoit en retour la liste triée selon le critère. (Plus exactement, le client envoie une liste de descripteurs de groupes, pas d'adresses IP des pairs potentiels.) On notera que ce mode permet au serveur de connaître les pairs potentiels de ses clients. En revanche, c'est ce mode qui permet au client de faire le moins de calculs, ce qui est utile pour les machines aux ressources réduites.
— Recherche indépendante de la cible ("*Target-independant Query Mode*") désigne une requête Alto où le client télécharge tout ou partie des descripteurs de groupe que connait le serveur, avec leurs attributs, et fait le calcul du meilleur pair lui-même. Préservant la vie privée du client, elle lui impose davantage de calculs.

Les exigences sur le futur protocole figurent en section 3 et sont numérotées AR-n (AR pour "*Alto Requirment*"). Elles vont des plus évidentes (exigence AR-1 : les clients et serveurs Alto doivent parler le protocole Alto...) aux plus pointues. Voici les plus importantes, selon moi.

Le protocole devra gérer les descripteurs de groupe : types multiples (AR-3), au moins les types « préfixe IPv4 » et « préfixe IPv6 » (AR-5) et extensible (possitibilité d'ajouter de nouveaux types de descripteurs de groupe dans le futur, AR-6). Les types « préfixes IP » sont, à ce stade, les types recommandés. Donc, en deux mots, les pairs possibles seront identifiés par leur adresse (« j'hésite entre un pair en `2001:db8:1:2::af` et un pair en `192.0.2.43`, tu me recommandes lequel ? »). Voir aussi AR-7 et AR-8.

Le protocole devra permettre d'indiquer un critère de classement (AR-11) permettant de comparer les machines entre elles. Alto ne détermine **pas** comment on fait les calculs : il utilise le résultat (on peut tout imaginer, y compris des tables gérées à la main, par exemple en fonction du fait que les liens sont de transit ou de "*peering*"). Alto devra, là aussi, être extensible et permettre d'ajouter d'autres critères dans le futur.

Ces critères doivent être relativement stables (ne pas oublier qu'une fois un transfert de fichiers commencé, il peut durer de nombreuses heures) donc ne pas dépendre, par exemple, de l'état de congestion du réseau (AR-13). À propos de la congestion, les applications qui utiliseront Alto ne doivent pas s'en servir comme remède contre la congestion, elles doivent utiliser un protocole qui dispose de mécanismes d'évitement de la congestion, comme TCP (AR-14, et aussi AR-29).

Le critère de classement doit pouvoir être indiqué au serveur (en mode « recherche dépendante de la cible », car, dans l'autre mode, c'est le client Alto qui classe) afin qu'il l'utilise pour classer (AR-16).

Où va être situé le client Alto ? Le protocole ne l'impose pas et il y aura au moins deux positionnements possibles : directement dans l'application (le client BitTorrent, par exemple) ou bien chez un tiers qui fera une partie du travail pour le compte du client (le "*tracker*" BitTorrent, par exemple). Alto devra fonctionner dans les deux cas (AR-18 et AR-19).

Et qui va fournir ses informations au serveur Alto ? Une possibilité évidente est que ce soit le FAI, et qu'Alto devienne un service de base de l'accès Internet, comme le résolveur DNS et le serveur NTP. Mais les exigences AR-20 et AR-21 demandent que le protocole n'impose pas ce choix : il faut que des acteurs tiers puisse jouer le rôle de serveur Alto.

Le protocole doit aussi permettre la redistribution de l'information obtenue (AR-25), en indiquant des conditions de redistribution.

Comment va-t-on trouver son serveur Alto ? Il faudra décrire un mécanisme et AR-37 demande que ce mécanisme soit intégré aux protocoles existants comme PPP ou DHCP (comme c'est le cas pour découvrir un résolveur DNS).

La sécurité étant un enjeu important pour Alto (voir aussi la section 5), il faudra aussi un mécanisme d'authentification des serveurs (AR-40), des clients (AR-41), la possibilité de chiffrer (AR-42), et d'ajuster la taille des requêtes et réponses selon le niveau de confiance du client envers le serveur, ou réciproquement (AR-44). Un client devra donc pouvoir rester intentionnellement vague dans sa demande, au risque évidemment que les informations du serveur soient moins pertinentes.

Et la sécurité ? Outre le RFC 5693 qui couvrait déjà la sécurité d'Alto, la section 5 de notre RFC met l'accent sur le point le plus important, le risque de distribution d'informations sensibles.

Ce risque existe dans les deux sens, du client vers le serveur et du serveur vers le client. Le client ne souhaite pas forcément communiquer au serveur tout ce qu'il sait. Par exemple, lorsque la HADOPI espionne, un client Alto ne va pas dire à un serveur qu'il ne connait pas « je cherche une copie HD du dernier Disney ». Et, en sens inverse, un serveur Alto géré par un FAI n'est pas forcément enthousiaste à l'idée de donner des informations qui permettent de se faire une bonne idée de la configuration de son réseau, de l'identité des opérateurs avec qui il "*peere*" et autres informations confidentielles.

Même si le client ne donne pas d'informations aussi évidentes que dans l'exemple ci-dessus, un serveur malin peut déduire plein de choses des demandes que le client lui adresse (la liste des pairs potentiels, par exemple, est une information utile pour repérer les plus grosses sources de contenu en pair-à-pair).

Pour l'opérateur du serveur, le principal risque est en recherche indépendante de la cible, lorsque le client télécharge de grandes quantités d'information. Un serveur prudent ne dira donc pas tout. En outre, même si le serveur accepte de donner ces informations au client, il peut quand même s'inquiéter de ce qu'elles deviendront. Un tiers qui écoute le réseau ne va t-il pas mettre la main sur l'information confidentielle ? Et, même si l'information donnée à chaque client est limitée (par exemple, uniquement une partie des descripteurs de groupes que le serveur connait), ne risque t-on pas de voir des clients coopérer pour récupérer chacun un bout, avant de le mettre en commun ?

Le futur protocole Alto devra donc contenir des mesures pour faire face à ces problèmes : par exemple, permettre au serveur d'être délibérement incomplet ou imprécis (annoncer un préfixe IP plus général que le vrai, par exemple), authentifier clients et serveurs, chiffrer les communications contre l'écoute par TMG, etc. En revanche, le RFC écarte l'idée de DRM dans les données Alto, complexes et facilement contournables. Les acteurs Alto sont donc prévenus que les données échangées ne peuvent pas être 100 % confidentielles. Un client ou un serveur méchant pourra toujours en faire mauvais usage.

À noter que les intérêts du client et du serveur peuvent être contradictoires. Ainsi, les requêtes en mode indépendante de la cible préservent la vie privée du client (il ne dit rien sur ce qu'il va faire des données) mais oblige le servur à diffuser de l'information. Et, pour les requêtes en mode dépendant de la cible, c'est le contraire :le client se dévoile, le serveur ne prend pas de risque.