RFC 7342 : Practices for Scaling ARP and Neighbor Discovery (ND) in Large Data Centers

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 28 août 2014

Date de publication du RFC : Août 2014

https://www.bortzmeyer.org/7342.html

Les protocoles de résolution d'adresse IP en adresse MAC sur un réseau local, ARP pour IPv4 et ND pour IPv6, fonctionnent par diffusion. Le client crie à tout le monde « qui connaît 2001:db8:376:ab::23:c0f ? » et la machine qui se reconnaît répond. L'inconvénient de ce mécanisme est qu'il passe mal à l'échelle : lorsque des centaines de milliers de machines virtuelles sont sur le même réseau local dans un grand "*data center*", et crient toutes en même temps, le réseau va souffrir. Ce nouveau RFC ne propose pas de solution mais décrit les pratiques qui sont actuellement utilisées pour limiter les dégâts.

Le problème des protocoles ARP (RFC 826 [1]) et ND (RFC 4861) dans les grands "*data centers*" a été décrit en détail dans le RFC 6820. Le problème a toujours existé mais est devenu bien plus sérieux depuis qu'on pratique massivement la virtualisation et que le nombre de machines qui font de l'ARP ou du ND a explosé. Le désir de souplesse dans la gestion de ces machines fait qu'il est difficile d'architecturer le réseau en fonction de ce problème. Par exemple, faire de chaque machine physique un réseau limiterait la diffusion des requêtes ARP ou ND mais obligerait à changer l'adresse IP des machines virtuelles lorsqu'elles passent d'une machine physique à une autre, annulant ainsi cette souplesse que fournit la virtualisation.

Les trois principales conséquences de cette augmentation du trafic ARP/ND :
— Consommation de capacité réseau au détriment du trafic « utile » (bien sûr, avec les liens modernes à 10 Gb/s, c'est moins grave qu'avant),
— Augmentation de la charge de travail des routeurs, sans doute la conséquence la plus gênante aujourd'hui (cf. une étude de Merit <http://tools.ietf.org/html/draft-karir-armd-statistics>),

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc826.txt

— En IPv4, augmentation de la charge de **toutes** les machines car la diffusion est totale (IPv6 utilise le "*multicast*"). Là encore, l'augmentation des ressources de calcul des équipements réseau peut aider : un test en laboratoire indique que 2 000 messages ARP par seconde consomment 2 % du CPU d'un serveur.

Comme expliqué dans la section 3, le réseau interne d'un grand "*data center*" a en général une de ces trois architectures :

— Connectivité au niveau 3 (L3), c'est-à-dire qu'on fait du routage IP dans le "*data center*",
— Tout en niveau 2, le "*data center*" est un grand réseau L2,
— Virtualisation du réseau ("*overlay*").

Pour chacune de ces architectures, une section de notre RFC décrit les pratiques actuelles pour limiter l'effet des protocoles de résolution.

Dans le premier cas (réseau L3), c'est la section 4. Les réseaux utilisant le routage mettent un routeur IP dans chaque baie, voire un pour chaque machine physique. Gros avantage : la diffusion des messages ARP ou ND, qui ne va pas au-delà du premier routeur, est très limitée. Le problème décrit dans le RFC 6820 disparaît donc complètement. Inconvénient : on n'a plus aucune souplesse. Changer une VM de baie, voire de serveur physique dans la même baie, oblige à la changer d'adresse IP, ce qui va casser les sessions en cours, nécessiter une reconfiguration, etc. Cette architecture ne convient donc que lorsque le "*data center*" est assez statique, ou bien lorsque les services qui y tournent peuvent supporter ces changements d'adresses IP.

Variante sur cette architecture (non mentionnée dans le RFC mais que j'emprunte à Vincent Bernat), annoncer dans le protocole de routage interne une route par machine (un préfixe /32 en IPv4 et /128 en IPv6). Cela résout ce problème et permet de tout faire en L3, mais, si on a des centaines de milliers de machines, le protocole de routage va souffrir.

Deuxième architecture possible (section 5), le grand réseau L2 où toute diffusion va frapper toutes les machines. Cette fois, le problème des messages de diffusion ARP ou ND va se poser en grand. Les routeurs qui vont faire communiquer ce réseau L2 avec l'extérieur vont souffrir. Comment diminuer cette souffrance ? D'abord, pour le cas où une machine cherche à communiquer avec une machine externe et doit donc résoudre l'adresse IP du routeur en adresse MAC. Si cette adresse MAC n'est pas dans le cache de la machine, elle envoie une requête, et le routeur doit la traiter, ce qui se fait en général via le CPU généraliste du routeur, pas dans les circuits spécialisés.

Première solution : pré-charger l'adresse MAC du routeur dans toutes les machines (options `-s` et `-f` de la commande `arp` sur Unix). Deuxième solution, plus souple : que le routeur envoie des réponses ARP ou ND spontanément, sans attendre les requêtes (cf. RFC 5227). Ainsi, son adresse MAC sera quasiment toujours dans les caches. Cela marche très bien pour IPv4. Mais cela ne résout pas complètement le problème pour IPv6 : ce protocole impose l'envoi de requêtes au routeur, pour valider que la liaison avec celui-ci fonctionne (RFC 4861, section 7.3, notamment « "*Receipt of other Neighbor Discovery messages, such as Router Advertisements and Neighbor Advertisement with the Solicited flag set to zero, MUST NOT be treated as a reachability confirmation. Receipt of unsolicited messages only confirms the one-way path from the sender to the recipient node. In contrast, Neighbor Unreachability Detection requires that a node keep track of the reachability of the forward path to a neighbor from its perspective, not the neighbor's perspective.*" »). Malgré cela, notre RFC recommande que cette pratique soit utilisée pour les réseaux IPv4 et que l'on travaille à améliorer la situation pour IPv6, dans la ligne du RFC 7048.

Second cas dans ce grand réseau L2, celui du traitement du trafic entrant dans le réseau. Le routeur de bordure reçoit un paquet pour une machine interne. L'adresse MAC de cette dernière n'est pas dans le cache ARP ou ND du routeur. Il faut donc émettre une requête ARP ou ND, ce qui fait du travail pour le CPU et utilise de nombreux tampons d'attente. Première solution : limiter la quantité de requêtes par seconde. Celles en excès seront simplement abandonnées. Deuxième solution : en IPv4, le routeur peut également surveiller le trafic ARP des machines afin d'apprendre leurs adresses MAC et de remplir son

cache à l'avance. En IPv6, le trafic ND n'est pas toujours diffusé et donc pas toujours écoutable, et cette solution n'aide donc pas tellement. Notre RFC recommande cette deuxième solution, au moins pour IPv4, en notant que, s'il est possible que beaucoup de paquets soient destinés à des machines éteintes ou injoignables (et qui n'émettent donc jamais), il vaut mieux trouver d'autres solutions.

De même qu'on peut pré-configurer statiquement l'adresse MAC du routeur dans toutes les machines, on peut aussi pré-configurer statiquement les adresses MAC de toutes les machines dans les routeurs. C'est évidemment un cauchemar à gérer (ceci dit, ce n'est pas fait à la main pour chaque VM, mais typiquement à l'intérieur du programme de création des VM) et il n'existe pas de mécanisme standard pour cela. Le RFC suggère que l'IETF regarde si on peut augmenter des protocoles comme NETCONF pour faire ce genre d'opérations.

Autre approche, rendre la résolution ARP hiérarchique (voir RFC 1027 et les travaux ultérieurs comme cette proposition <https://tools.ietf.org/html/draft-shah-armd-arp-reduction>). Au lieu de n'avoir que deux niveaux, le client et le serveur, on pourrait avoir un relais ARP spécialisé qui puisse répondre aux requêtes, garder en cache les résultats, etc. Le RFC 4903 déconseillait cette approche, notamment parce qu'elle peut gêner le déplacement rapide d'une machine (il faut alors invalider les caches) et parce qu'elle est incompatible avec SEND (que personne n'utilise, il est vrai, cf. RFC 3971). Même possibilité, au moins en théorie, avec ND (RFC 4389 décrit un relais mais qui ne réduit pas le nombre de messages). Au bout du compte, notre RFC ne recommande pas cette technique, bien qu'elle ait parfois été déployée.

Troisième architecture possible pour un grand "*data center*", les réseaux virtuels ("*overlays*", section 6). Il existe des normes pour cela (TRILL ou IEEE 802.1ah) et plusieurs projets. L'idée est que seul un petit nombre d'adresses MAC sont visibles dans les paquets qui circulent entre les baies, les paquets étant décapsulés (révélant l'adresse MAC interne) à un endroit très proche des VM. Les équipements qui font l'encapsulation et la décapsulation doivent être suffisamment dimensionnés puisqu'ils se taperont presque tout le trafic ARP et ND. L'intensité du travail peut être diminuée par des correspondances statiques (comme cité pour les autres architectures) ou bien en répartissant astucieusement la charge entre les machines d'encapsulation/décapsulation.

La section 7 du RFC résume les recommandations : pas de solution miracle qui conviendrait à tous les cas, plusieurs solutions mais aucune parfaite. Certaines améliorations nécessiteraient un changement des protocoles et cela fait donc du travail possible pour l'IETF :

— Atténuer certaines obligations de ND concernant la bidirectionnalité, obligations qui interdisent actuellement de profiter complètement des messages ND spontanés.
— Spécifier rigoureusement les sémantiques des mises à jour (lorsqu'on reçoit une réponse ARP ou ND alors qu'on a déjà l'information en cache).
— Développer des normes pour des relais/caches pouvant stocker les réponses ARP ou ND et les redistribuer largement. Le RFC 4389 peut donner des idées.

À l'heure actuelle, il n'y a plus de groupe de travail IETF sur ce sujet et ce développement de nouvelles solutions risque donc de ne pas être immédiat.