RFC 6820 : Address Resolution Problems in Large Data Center Network

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 30 janvier 2013

Date de publication du RFC : Janvier 2013

https://www.bortzmeyer.org/6820.html

Lorsque des dizaines de milliers, voire des centaines de milliers de machines, sont connectées au même réseau (ce qui peut être le cas dans un grand "*data center*"), certains protocoles TCP/IP, pas forcément prévus pour cela, peuvent avoir des problèmes. C'est le cas d'ARP et de NDP, protocoles utilisés pour trouver l'adresse MAC d'une machine voisine. ARP n'avait pas été prévu pour les "*data centers*" modernes et ses limites se font parfois sentir. ND pose pas mal de problèmes similaires. D'où ce document qui définit précisement le problème.

Il n'y aura par contre pas de solution de si tôt : le groupe de travail armd <http://tools.ietf.org/wg/armd>, qui devait travailler sur ce sujet, a été dissous immédiatement après avoir produit ce RFC de description du problème. Il ne semblait pas qu'il soit possible de travailler sur des solutions et l'IESG a donc choisi de dissoudre le groupe <http://www.ietf.org/mail-archive/web/armd/current/msg00472.html> alors qu'il était loin du résultat. Il reste cet intéressant RFC 6820 [1] qui sera lu avec plaisir par tous les passionnés de réseaux informatiques. (Un autre document, le RFC 7342, décrit les pratiques existantes.)

ARP (RFC 826) et ND (RFC 4861) s'attaquent tous les deux (le premier pour IPv4 et le second pour IPv6) au problème de la résolution d'adresses sur le réseau local. Une machine veut parler à une autre machine dont l'adresse IP indique qu'elle est sur le même réseau local. Pas besoin de routage, donc, on peut lui envoyer directement un paquet mais, pour cela, il faut d'abord trouver son adresse MAC. ARP et ND procèdent de façon très proche : ils diffusent un paquet à tout le monde « qui est 2001:db8:1::af:41 ? » et la machine qui se reconnait répond. Cela veut dire que ce paquet de demande va être transmis à toutes les machines du réseau local, jusqu'aux routeurs qui marquent la limite du « domaine de "*broadcast*" » (l'ensemble des machines touchées par la diffusion « L2 », c'est-à-dire

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc6820.txt

utilisant les mécanismes de la couche 2). Des détails comme les VLAN compliquent encore le tableau. Pour notre RFC, un « domaine de "*broadcast*" L2 » est l'ensemble des liens, répéteurs et commutateurs traversés pour atteindre tous les nœuds qu'on touche avec une diffusion.

(Il existe une légère différence entre ARP et ND, ND utilisant le "*multicast*" et pas le "*broadcast*" mais, en pratique, cela ne change pas grand'chose : ce point est traité plus loin.)

Depuis longtemps, on sait que les grands domaines de diffusion (réseaux « plats », sans routeurs) créent des tas de problèmes. La totalité des machines du réseau doit traiter les paquets de diffusion. Une machine méchante ou détraquée qui envoie des paquets de diffusion en boucle peut sérieusement perturber un réseau. Dans certains cas (« tempêtes de diffusion », provoquées par exemple par une boucle non détectée dans la topologie du réseau), ces paquets diffusés suscitent la création d'autres paquets diffusés et, là, on peut arrêter complètement le réseau. La sagesse des administrateurs réseaux est donc depuis longtemps « ne faites pas de grands domaines de diffusion ; partitionnez en mettant des routeurs au milieu »). Par exemple, si on architecture son "*data center*" en mettant un sous-réseau IP par armoire, la charge résultant d'ARP et de ND sera confinée à une seule armoire. Et les éventuelles tempêtes de diffusion resteront dans une armoire. Le problème est que cette configuration est plus rigide : avec les techniques de gestion modernes ("*buzzwords*" : "*cloud*", élasticité, agilité, etc), on souhaite pouvoir déplacer une machine très facilement, sans la renuméroter. Les grands réseaux L2 plats permettent cela. Au contraire, les réseaux très partitionnés nécessitent de changer l'adresse IP de la machine selon sa position physique dans le "*data center*", faisant perdre en souplesse de gestion. Même problème lorsqu'on veut ajouter des nouvelles machines : avec un réseau plat, on configure les machines et on les installe là où il y a de la place. Avec un réseau très partitionné, on doit les configurer en fonction de l'endroit où elles vont être installées, ajoutant ainsi une contrainte de gestion.

Et la virtualisation, très utilisée dans les "*data centers*" actuels, pousse encore plus dans ce sens. Avec la virtualisation, déplacer une machine ne nécessite même plus d'opérations physiques : on copie la VM, c'est tout. Si l'alimentation électrique d'une machine physique menace de défaillir, on copie toutes les VM hébergées dans l'armoire vers d'autres machines physiques, puis on peut arrêter l'armoire tranquillement. Une architecture où chaque armoire est son propre sous-réseau ne permet plus cela : changer d'armoire veut dire changer d'adresse IP, donc sérieusement impacter le service.

Autre conséquence de la virtualisation : le nombre de machines augmente. Il est courant aujourd'hui d'avoir plusieurs dizaines de machines virtuelles par machine physique. Demain, ce nombre sera probablement plus élevé. Les problèmes de passage à l'échelle d'ARP et de ND vont donc devenir plus aigus lorsqu'une VM émettra un paquet ARP qui devra être traité par des milions de machines situées dans le même "*data center*" (des "*data centers*" de 100 000 machines physiques existent déjà, ce qui fait bien davantage de VM).

Le désir de souplesse et d'agilité dans la gestion des machines virtuelles (VM) pèse aussi sur l'architecture du réseau : il est plus difficile de prédire les flux de données entre machines (ça change tout le temps) et on ne peut donc pas architecturer le réseau autour des flux d'aujourd'hui, par exemple pour conserver le trafic à l'intérieur d'une seule armoire, ou pour éviter de passer par plusieurs commutateurs.

Les sections 4 et 5 se penchent sur les spécificités des deux protocoles ARP et ND. ARP souffre du fait d'être un très vieux protocole, nettement sous-spécifié (par exemple, aucun RFC ne précise de stratégie de retransmission lorsqu'un paquet est perdu). Les mises en œuvre d'ARP sont très variables dans leur comportement. Certaines sont par exemple bien plus bavardes que d'autres, vidant les caches même lorsque ce n'est pas nécessaire, ce qui entraîne davantage de requêtes.

ND est mieux placé qu'ARP, car plus récent. Ceci dit, pour la plupart des problèmes listés dans ce RFC, ND et ARP sont proches. La principale différence est qu'ARP utilise le "*broadcast*" et ND le "*multicast*". En théorie, cela permet que les requêtes n'aillent pas dans tout le réseau, seulement dans les parties où il y a des machines IPv6. En pratique, je suis sceptique : combien de commutateurs filtrent ainsi ? Mais, même si cela ne protège pas le réseau, cela protège les machines purement IPv4, qui peuvent ignorer ces sollicitations dès l'arrivée sur la carte Ethernet. Évidemment, si toutes les machines parlent IPv6, cet avantage disparait : tout le monde droit traiter les requêtes ND, comme tout le monde droit traiter les requêtes ARP.

Dernière chose à garder en tête avant de voir la liste des problèmes, l'architecture du "*data center*" (section 6). Un "*data center*" typique est hiérarchique : des armoires, connectées à un réseau d'accès, lui-même connecté à un réseau d'agrégation, lui-même connecté au cœur. L'armoire compte plein de serveurs et le réseau d'accès est, soit un commutateur réseau interne (ToR pour "*Top of Rack*" car il est souvent en haut de la baie), soit on utilise un commutateur situé à l'extrémité de la rangée d'armoirees (EoR pour "*End of Row*").

Ensuite, le réseau d'agrégation qui, dans le cas le plus fréquent, est composé de commutateurs connectant les commutateurs ToR (souvent plusieurs centaines) entre eux.

Le cœur, lui, comprend les équipements qui connectent les commutateurs du réseau d'agrégation, ainsi que les liens avec l'extérieur.

Maintenant, la question d'architecture qui va le plus impacter notre problème : jusqu'où va le domaine de diffusion d'un serveur ? Jusqu'au réseau d'accès, jusqu'au réseau d'agrégation, ou bien jusqu'au cœur ? Comme vu plus haut, la première solution est plus sûre (les problèmes seront concentrés dans une armoire ou dans un relativement petit groupe d'armoires), la seconde plus flexible et la troisième offre le maximum de flexibilité... et de risques.

Pour profiter de la structure hiérarchique du "*data center*", on souhaite que le trafic entre deux machines connectées au même réseau d'accès reste local et n'aille pas sur le réseau d'agrégation. Idem entre l'agrégation et le cœur. Si le "*data center*" est occupé par une seule organisation, où le trafic réseau est bien connu et compris, cela peut être possible. Par exemple, si on héberge une ferme de serveurs Web dynamiques, dépendant d'un SGBD, on prendra soin de mettre les serveurs de base de données proches du frontal HTTP, puisqu'ils auront beaucoup à parler entre eux.

Si le "*data center*" est public (utilisé par des tas de clients différents, n'ayant pas de lien entre eux, par exemple Amazon EC2, OVH ou Gandi), on ne peut plus espérer connaître les flux réseau. Bref, le type d'usage envisagé va peser sur l'architecture.

Enfin, après ces préliminaires, la section 7 expose la liste des problèmes ARP/ND identifiés. D'abord, le traitement par les routeurs. Ceux-ci doivent écouter le trafic ARP et traiter chaque paquet, ne serait-ce que pour mettre le cache ARP à jour (comme l'impose le RFC 826). Ce traitement se fait en général sur le « chemin lent » ("*slow path*") du routeur, c'est-à-dire en remontant au CPU généraliste de la machine, sans pouvoir utiliser les plus rapides ASIC dont on se sert pour la transmisison de paquets. Ce CPU est limité et, avec les machines actuelles, il ne faut pas espérer plus de quelques milliers de paquets ARP par seconde, ce qui peut être insuffisant dans certains environnements.

Une des techniques utilisées par les routeurs est d'ignorer très tôt les requêtes ARP où l'adresse du routeur n'apparait pas. Cela viole le RFC 826 mais cela peut être nécessaire.

Une autre optimisation est de limiter le nombre de requêtes qui concernent le routeur en envoyant périodiquement de l'ARP gratuit (RFC 5227), pour garder les caches des autres machines peuplés (à noter que cela ne marche pas si la machine n'avait pas déjà une entrée dans son cache ARP).

Autre problème pour le routeur, chaque fois qu'on doit envoyer un paquet à une machine pour laquelle le routeur n'a **pas** d'entrée dans son cache ARP, il faut effectuer un traitement assez lourd (mettre le paquet IP en attente, comme imposé par les RFC 1122 et RFC 1812, envoyer la requête ARP, attendre et réémettre si nécessaire, peut être envoyer une réponse ICMP "*destination unreachable*") et qui est typiquement géré par le "*slow path*" du routeur, sa partie la plus lente. Si le "*data center*" contient un grand réseau à plat, et que le routeur doit parler à des dizaines de milliers de machines (et donc faire de l'ARP avec elles), cela peut être une charge sérieuse. (Ce n'est pas le nombre d'interfaces physiques du routeur qui compte mais le nombre totale de machines avec qui il parle en couche 2.)

Ensuite, le cas du protocole ND d'IPv6. Il est très proche d'ARP et bien des problèmes sont les mêmes. La principale différence est qu'il utilise le "*multicast*" et que donc, dans certaines circonstances, ses paquets pourront être ignorés, diminuant la charge sur les équipements réseau.

Mais, par contre, le RFC 4861, section 7.3.1, impose des vérifications régulières de la validité des entrées dans le cache. La méthode de vérification n'est pas spécifiée (on a le droit de se servir des accusés de réception TCP pour cela). Mais cela pousse certaines machines à envoyer des demandes ND répétées, parfois une toutes les trente-cinq secondes. Et il n'y a pas d'équivalent de l'ARP gratuit, le RFC 4861, section 7.3.1, disant que la réception d'une réponse ND ne doit pas valider une entrée du cache (au contraire, cela peut générer un test ND, précisement ce qu'on tentait d'éviter). C'est parce que la réception d'une « réponse » ne prouve que le chemin retour, pas forcément l'aller.

La plupart des réseaux étant aujourd'hui en IPv4 **et** IPv6, les problèmes quantitatifs s'additionnent. La situation ne s'arrangera que lorsqu'on pourra arrêter IPv4, ce qui est une perspective encore très éloignée.

Derniers équipements à souffrir (le RFC ne considère pas le cas des machines terminales), les commutateurs. Pour n'envoyer les paquets que sur le bon port physique (au lieu de le diffuser partout), le commutateur doit garder en mémoire une table des adresses MAC. Si le réseau « couche 2 » est très grand, cette table le sera aussi.

Si vous voulez approfondir le sujet et notamment voir toutes les données quantitatives, lisez le document draft-karir-armd-statistics <http://tools.ietf.org/id/draft-karir-armd-statistics>, et le RFC 7342.