RFC 7948 : Internet Exchange BGP Route Server Operations

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 9 septembre 2016

Date de publication du RFC : Septembre 2016

https://www.bortzmeyer.org/7948.html

Les points d'échange Internet sont un des maillons essentiels du bon fonctionnement de l'Internet, permettant des connexions faciles et relativement bon marché entre opérateurs. Une fois physiquement connecté au point d'échange, l'opérateur Internet doit encore établir une connexion BGP (RFC 4271 [1]) avec ses pairs. Si ceux-ci sont nombreux, établir N connexions bilatérales représente un coût administratif important. Il est souvent préférable d'établir une seule connexion multilatérale, par l'intermédiaire d'un **serveur de routes** ("*route server*"). Celui-ci récolte les annonces BGP de tous ses pairs et les redistribue. Ainsi, il suffit à l'opérateur d'une seule connexion, avec le serveur de routes. Ce nouveau RFC documente les questions opérationnelles liées aux serveurs de routes.

Le principe et le fonctionnement d'un serveur de routes sont expliqués dans le RFC 7947, produit par un autre groupe de travail IETF. Ce RFC 7948 se consacre aux détails opérationnels. Un serveur de routes est l'équivalent, pour le BGP externe (eBGP) de ce qu'est un « réflecteur de routes » ("*route reflector*", RFC 4456) pour le BGP interne (iBGP). Le serveur de routes, lorsqu'il existe, est un composant crucial du point d'échange. Par exemple, s'il tombe en panne, les sessions BGP sont coupées, les annonces retirées et, même si les commutateurs et le réseau du point d'échange continuent à fonctionner, plus aucun trafic ne passe (à part si des sessions BGP bilatérales existent également). D'où l'importance d'une bonne gestion de ce composant.

Les sections 2 et 3 de notre RFC rappelent la différence entre sessions BGP bilatérales et multilatérales à un point d'échange. Si on ne fait que des sessions bilatérales (pas de serveur de routes), avec seulement quatre routeurs BGP sur le point d'échange, il faudra six sessions BGP pour une connectivité complète. Avec dix routeurs, il en faudrait quarante-cinq ! Établir, superviser et maintenir toutes ces sessions représente du travail. Les sessions multilatérales, via un serveur de routes, sont une bien meilleure

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc4271.txt

solution. Avec dix routeurs au point d'échange, il n'y a plus besoin que de dix sessions BGP, chacun des dix routeurs ne faisant que BGP qu'avec le serveur de routes.

Le serveur de routes doit juste veiller à ne pas toucher à l'attribut BGP `NEXT_HOP` (RFC 4271, section 5.1.3), qui ne doit pas indiquer le serveur de routes, mais le routeur qui a annoncé la route (le serveur de routes n'est **pas** un routeur, et ne fait pas de transmission du trafic IP). Pour le point d'échange typique, il n'y a pas besoin de faire de la résolution récursive (utiliser un protocole de routage pour trouver comment joindre le routeur suivant) du `NEXT_HOP` : tous les routeurs sont sur le même réseau L2 et sont donc joignables directement. Par exemple, si je regarde le looking glass du France-IX `<http://lg.franceix.net/>` et que lui demande les routes vers `194.0.9.0/24`, il me montre :

```
194.0.9.0/24       via 37.49.236.20 on eth1 [RS1_PAR 2016-08-11 from 37.49.236.250] * (100) [AS2484i]
                   via 37.49.236.21 on eth1 [RS1_PAR 2016-08-11 from 37.49.236.250] (100) [AS2486i]
...
```

(L'AFNIC est connectée sur deux points de présence de ce point d'échange, d'où les deux routeurs.) Les `NEXT_HOP` sont les adresses des routeurs dans le point d'échange, `37.49.236.0/23`.

La section 4, le gros morceau de notre RFC, décrit ensuite divers points que le serveur de routes doit garder en tête, pour faire un bon travail.

D'abord, le cas où on n'envoie pas exactement les mêmes informations à tous les clients. Sauf si le client BGP coopère, cela implique que le serveur garde une base des routes par groupe de clients, un groupe étant composé de tous les clients pour qui on applique la même politique de filtrage. Attention, cela consomme pas mal de mémoire (autant de base que de groupes) et de processeur (il faut faire tourner les algorithmes de sélection BGP pour tous les groupes) mais heureusement, en pratique, on utilise rarement ces politiques différentes. Traiter tous les clients de la même façon permet de garder une consommation de ressources raisonnable.

Qu'en est-il du risque de fuite de préfixes ? Si un routeur connecté au serveur de routes fuit et envoie, par exemple, la totalité de la DFZ au serveur de routes, celui-ci va-t-il transmettre tous ces préfixes à ses infortunés clients ? Cela serait très ennuyeux car tout le trafic partirait alors vers le routeur fautif, qui ne serait peut-être pas capable de le gérer. Il est donc très recommandé que le serveur de routes prenne des précautions contre les fuites. Au minimum, imposer un nombre maximal de préfixes à chaque client, et, idéalement, filtrer les préfixes autorisés pour chaque client. C'est moins grossier que la simple limite quantitative, mais c'est aussi plus dur à maintenir (on ne peut pas espérer que tous les clients tiennent à jour leurs préfixes dans les IRR...). Il est certainement préférable que les administrateurs des clients et du serveur de routes regardent le journal de leur routeur pour repérer les différences entre ce qui est théoriquement annoncé et ce qu'il l'est réellement.

Question fiabilité, notre RFC recommande que le serveur de routes soit redondant : s'il n'est composé que d'une seule machine, et qu'elle plante, le point d'échange ne servira plus. Il faut donc au moins deux machines, et prendre soin qu'elles soient configurées de manière à annoncer les mêmes routes. (C'est trivial si les machines sont identiques, il suffit qu'elles aient la même configuration, cela l'est moins si, pour augmenter la redondance, on choisit des machines ayant des logiciels différents.)

Autre précaution à prendre, côté client, ne pas vérifier que le chemin d'AS est cohérent. Par exemple, le serveur de routes ne va pas mettre son propre AS tout à gauche du chemin d'AS (RFC 7947, section 2.2.2) et le client ne doit donc pas vérifier que son pair, le serveur de routes, a un chemin d'AS incluant l'AS du pair BGP.

Comment contrôler l'exportation des routes vers certains clients ? La section 4.6 liste plusieurs méthodes :

— Demander aux clients d'étiqueter les routes avec une communauté ordinaire (RFC 1997) ou bien une communauté étendue (RFC 4360). Attention, la taille limitée des valeurs des communautés ne permet pas d'y mettre deux numéros d'AS si ceux-ci sont sur quatre octets (RFC 4893). Comme exemple, regardez la politique des communautés BGP au France-IX <https://apps.db.ripe.net/search/query.html?full_query_string=&searchtext=AS51706&do_search=Search&inverse_attributes=None&ip_search_lvl=&alt_database=RIPE&object_type=aut-num&object_template=none&recursive=ON#resultsAnchor>.
— Utiliser les politiques de distribution exprimées en RPSL (RFC 2622) dans les IRR.
— Utiliser un IRR local au point d'échange, avec son interface spécifique pour y enregistrer des politiques. Cette posibilité est mentionnée par le RFC, mais quelqu'un connait un point d'échange qui fonctionne comme cela ? (Peut-être VIX et AMS-IX, à vérifier.)

On a vu que la grande majorité des points d'échange travaillaient au niveau 2. Normalement, toute machine connectée au réseau du point d'échange peut donc joindre n'importe quelle autre. Mais, en pratique, on a déjà vu des bogues dans un commutateur qui menaient à des communications non transitives. Par exemple, les routeurs A et B peuvent tous les deux joindre le serveur de routes mais ne peuvent pas se joindre mutuellement. Dans ce cas, A reçoit bien les routes de B (et réciproquement) mais, lorsqu'il essaie de transmettre des paquets à B, ceux-ci finissent dans un trou noir. Ce problème est spécifique aux serveurs de route : lorsqu'on a une connexion bilatérale, les paquets de contrôle (ceux envoyés en BGP) suivent en général le même chemin que ceux de données (les paquets IP transmis) et partagent donc le même sort, bon ou mauvais.

La solution à ce problème passe peut-être par des solutions comme BFD (RFC 5881), pour tester la connectivité entre les deux routeurs. Mais BFD nécessite une configuration bilatérale entre les deux routeurs et il n'existe actuellement aucun protocole pour faire cette configuration automatiquement. On retombe donc dans les problèmes de configuration manuelle d'un grand nombre de connexions bilatérales. Même si un protocole de configuration automatique existait, il resterait le problème d'informer le serveur de routes. En effet, un échec BFD indiquerait à A qu'il ne peut pas joindre B, et lui ferait marquer les routes vers B comme invalides mais le serveur de routes, n'étant pas au courant, continuerait à n'envoyer à A que la route invalide (un pair BGP ne transmet que la meilleure route vers une destination donnée).

Dernier piège, le détournement de `NEXT_HOP`. On a vu que le fonctionnement normal d'un serveur de routes est de ne pas se mettre comme « routeur suivant » mais au contraire de relayer aveuglément les `NEXT_HOP` indiqués en BGP par ses clients. Un client malveillant pourrait annoncer des routes au serveur de routes en indiquant comme « routeur suivant » le routeur d'un concurrent, pour lui faire acheminer son trafic, par exemple ou, pire, pour faire une attaque par déni de service en annonçant des préfixes très populaires avec le concurrent comme routeur suivant. Contrairement au problème précédent, celui-ci n'est pas spécifique aux serveurs de route, il existe aussi avec des sessions BGP bilatérales. Mais il peut faire davantage de dégâts lorsqu'on utilise un serveur de routes, tous les clients croyant l'annonce mensongère. Pour empêcher cela, il faudrait que les serveurs de route vérifient que l'attribut BGP `NEXT_HOP` corresponde à l'adresse IP du client. (Les clients ne peuvent pas faire ce test, ils doivent croire le serveur de routes, qui annonce des `NEXT_HOP` qui ne sont pas son adresse IP.)

Et pour finir, une sortie complète du "*looking glass*" du "*route server*" du France-IX concernant un serveur racine du DNS, `M.root-servers.net` (et, au passage, attention en anglais à ne pas confondre "*route server*" et "*root server*") :

```
RS1 show route for 202.12.27.33/32 all - View the BGP map

202.12.27.0/24     via 37.49.237.106 on eth1 [RS1 2015-08-21 from 37.49.236.250] * (100) [AS7500i]
Type: BGP unicast univ
BGP.origin: IGP
BGP.as_path: 7500
BGP.next_hop: 37.49.237.106
```

```
BGP.med: 500
BGP.local_pref: 100
BGP.atomic_aggr:
BGP.aggregator: 192.50.45.1 AS7500
BGP.community: (51706,51706) (51706,64601) (51706,64650)
```

Les pages de présentation de quelques serveurs de route :
— Celle de Netnod <http://www.netnod.se/ix/routeservers>,
— Celle d'AMS-IX <https://ams-ix.net/technical/specifications-descriptions/ams-ix-rout avec des exemples de configuration pour certains routeurs,
— Et l'exposé d'Arnaud Fenioux au FRnog <http://media.frnog.org/FRnOG_25/FRnOG_25-3.pdf>.

Merci à Arnaud Fenioux pour sa relecture et ses ajouts et corrections.