RFC 7606 : Revised Error Handling for BGP UPDATE Messages

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 28 août 2015

Date de publication du RFC : Août 2015

https://www.bortzmeyer.org/7606.html

Que doit faire un routeur BGP lorsqu'il reçoit un message de type UPDATE avec un attribut incorrect ? La norme était claire : le routeur doit fermer la session BGP en cours (et donc perdre toutes les routes associées). Cela partait du bon sens (si l'attribut est corrompu, on ne peut pas se fier au routeur qui l'a envoyé) mais cela avait des conséquences sérieuses : on supprimait toutes les routes, pas seulement celle dans l'annonce UPDATE. Ce court RFC modifie donc BGP sur un point : on ne coupe plus forcément toute la session, on retire uniquement la route qui figurait dans l'annonce incorrecte.

L'ancienne norme figurait dans le RFC 4271 [1], section 6 : « "*When any of the conditions described here are detected, a NOTIFICATION message, with the indicated Error Code, Error Subcode, and Data fields, is sent, and the BGP connection is closed*" ». En pratique, cela voulait dire que les routeurs coupaient des sessions simplement à cause d'un attribut mal formé. Cela pose un problème de sécurité : comme certains routeurs ne vérifient pas les attributs des annonces, l'annonce avec l'attribut invalide peut être propagée et planter des sessions situées bien après l'origine de l'annonce, rendant le déboguage et l'attribution des responsabilités très difficiles. En outre, l'annonce a pu être dupliquée par ces routeurs qui ne vérifient pas, et une seule annonce peut donc planter plusieurs sessions. Cela s'était produit, par exemple, dans le cas du fameux attribut 99 <https://www.bortzmeyer.org/bgp-attribut-99.html>.

Que peut faire un routeur BGP lorsque il reçoit une annonce invalide ? La section 2 du RFC liste les possibilités, de la plus violente à la plus modérée :

— Réinitialiser la session (« dans le doute, reboote »). C'était l'approche officielle avant ce RFC.
— Ne réinitialiser qu'une seule famille d'adresses (AFI, "*Address Family Identifier*", comme seulement IPv4 ou seulement IPv6), comme documenté dans le RFC 4760.

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

— Considérer que l'annonce invalide est équivalente à un retrait des routes qu'elle contient ("*treat-as-withdraw*"). C'est une nouveauté de ce RFC, qui n'existait pas dans BGP avant. Cela évite de perdre toutes les routes de la session, comme c'était le cas avec la première approche.
— Ignorer l'attribut invalide mais garder l'annonce, ce que notre RFC déconseille formellement (sauf si l'attribut n'avait aucune conséquence sur la sélection et l'installation des routes).

L'approche « ignorer l'annonce invalide » n'est **pas** citée : dans un protocole où les mises à jour sont incrémentales, comme BGP (qui n'envoie pas la table de routage complète, seulement les changements), elle pourrait mener à des routes impossibles à détruire (cf. section 6).

C'est la section 3 qui contient les nouvelles règles exactes, après moultes discussions à l'IETF. Pour résumer : c'est la troisième option ("*treat-as-withdraw*") qui est désormais recommandée dans la plupart des cas.

Le reste du RFC est consacré à des détails pratiques. Par exemple, en section 5, on trouve des règles d'encodage qui permettront d'accéder aux routes annoncées (NLRI, "*Network Layer Reachability Information*") malgré la présence d'attributs mal formés. En effet, c'est très joli de dire qu'on doit traiter une annonce invalide comme un retrait mais il faut pour cela savoir quelles routes retirer (réinitialiser toute la session est bien plus simple à mettre en œuvre). Quand l'annonce est invalide, l'analyser n'est pas trivial. Notre RFC demande donc de faciliter la tâche du routeur de réception de l'annonce, par exemple en encodant les attributs `MP_REACH_NLRI` et `MP_UNREACH_NLRI` au tout début de la liste des attributs (pour pouvoir les comprendre même si l'annonce est invalide). Évidemment, les routeurs anciens ne suivent pas forcément ces règles et les récepteurs doivent donc rester prêts à tout.

Bien sûr, rien n'est parfait. L'ancienne règle de couper toute la session n'était pas due au désir des auteurs de BGP de perturber le plus possible l'Internet. Il y avait de bonnes raisons à cette décision, notamment de garantir la cohérence du routage. Avec la nouvelle règle, ce n'est plus aussi simple et on risque donc des tables de routage incohérentes (un routeur ayant accepté l'annonce et un autre l'ayant traité comme un retrait...), avec leurs conséquences, comme des boucles de routage. Cela explique la très longue gestation de ce RFC, due à de nombreuses discussions à l'IETF. Il faut dire que toucher à BGP est toujours délicat : une erreur peut potentiellement planter tout l'Internet.

La section 7 du RFC décrit en détail ce que veut dire « malformé » pour un attribut BGP. Par exemple, l'attribut `ORIGIN` (RFC 4271, section 4.3, et qui indique la source de l'information contenue dans l'annonce) a normalement une longueur de 1 (les attributs BGP sont encodés en TLV) et toute autre longueur indique un attribut `ORIGIN` mal formé : autrefois, cela aurait coupé la session, depuis notre RFC, cela doit entrainer un retrait de la route contenue dans l'annonce. Pour l'attribut `ORIGIN`, même chose si la valeur de l'attribut n'est pas une des valeurs spécifiées (`IGP`, `EGP` ou `INCOMPLETE`).

Autre exemple, l'attribut `COMMUNITIES` (RFC 1997) doit avoir une longueur qui est un multiple de 4. Si ce n'est pas le cas =¿ attribut mal formé =¿ annonce traitée comme étant un retrait de routes.

Conséquence de ce nouveau RFC : tout nouvel attribut spécifié doit indiquer le traitement à appliquer en cas de malformation (section 8). Ce sera en général "*treat-as-withdraw*" mais cela doit être marqué explicitement dans la norme décrivant le nouvel attribut.

Un avantage du long délai avant la sortie de ce RFC, est que ce nouveau comportement a déjà été mis en œuvre dans la plupart des routeurs (Alcatel-Lucent SR OS, Cisco IOS, Cisco IOS XR, Juniper JUNOS, Quagga).