RFC 10063 : Near Real Time Mirroring (NRTM) version 4

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 17 novembre 2026

Date de publication du RFC : Novembre 2026

https://www.bortzmeyer.org/10063.html

Ce RFC normalise enfin le protocole NRTM ("*Near Real Time Mirroring*"), qui permet de synchroniser les IRR, les registres de routes Internet. Ce protocole joue donc un grand rôle dans la sécurité du routage.

Le titre annonce NRTM version 4. Ne cherchez pas les RFC des versions 1, 2 et 3, ils n'ont jamais été écrits (ce qui n'a pas empêché la version 3 d'être largement déployée <https://docs.db.ripe.net/RIPE-Database-Mirror/Access-to-NRTM/>). Mais cette fois, on a une norme technique correcte pour le protocole NRTM. À quoi sert-il ? Pour le comprendre, il faut expliquer les IRR ("*Internet Routing Registries*" <https://irr.net/>) et pour comprendre leur utilité, il faut revenir sur la façon dont se fait le routage sur l'Internet. (Je vous recommande la lecture du RFC 7682[1] pour une vision générale des IRR, ainsi qu'une visite de .)

L'Internet est un réseau de réseaux, décentralisé. Chacun des réseaux qui composent l'Internet détient un certain nombre de préfixes d'adresses IP. Par exemple, Gitoyen a les préfixes 80.67.160.0/19 et 2001:910::/32. Les routeurs de chaque réseau vont annoncer aux routeurs des réseaux voisins les préfixes qu'ils savent joindre. Cela peut être transitif : un réseau va annoncer non seulement ses propres préfixes mais aussi ceux des réseaux qu'il sait joindre. Cela dépend de la politique de routage de chaque réseau. Mais vos voyez se poindre un problème de sécurité : comment savoir si une annonce est légitime ? Si l'opérateur Gitoyen annonce une route vers le préfixe 2a0f:6780::/29, comment savoir s'il a le droit de le faire, ou bien s'il est en train d'essayer de capter méchamment le trafic à destination de ce réseau, chose qui se produit parfois <https://www.bortzmeyer.org/pakistan-pirate-youtube.html> ? Il existe plusieurs techniques pour diminuer ce risque et l'une des plus anciennes est le registre de routes, ou IRR ("*Internet Routing Registry*").

Dans ces registres, dont l'accès est authentifié, on stocke les routes légitimes, décrites dans le langage RPSL (RFC 2622) et indiquant notamment l'AS d'origine. Lorsqu'un routeur reçoit en BGP une annonce de route, il peut vérifier dans le registre si elle semble légitime. Bon, j'ai simplifié : dans la réalité, le routeur ne regarde pas en temps réel, on utilise plutôt les IRR pour construire une liste de préfixes acceptés et cette liste est chargée dans la mémoire du routeur. Voici, vu avec whois, un exemple d'objet de type route au RIPE, montrant que l'AS 207666 est autorisé comme origine du préfixe :

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc7682.txt

```
% whois 2a0f:6780::/29
...
route6:          2a0f:6780::/29
descr:           Route to Gitoyen IPv6
origin:          AS20766
mnt-by:          Gitoyen-NCC
created:         2020-08-14T10:34:23Z
last-modified:   2020-08-14T10:34:23Z
```

Les registres de routes sont de grande taille (il y a actuellement dans les 1,4 millions de routes sur la DFZ) et si on veut les copier, il faut pouvoir le faire de manière efficace, d'autant plus qu'ils changent en permanence. En outre, il existe de nombreux IRR (au moins une douzaine de gros IRR connus actuellement, gérés par des organisations différentes, pas seulement les RIR), et qui souhaitent parfois enrichir leur base avec les données d'autres IRR. Se copier les uns les autres est classique pour les IRR. Un mécanisme de synchronisation est donc nécessaire, et s'il est standard, c'est encore mieux.

C'est précisément ce que fournit NRTM. Le RFC note d'ailleurs que ces IRR ne servent pas qu'au filtrage d'annonces de routes mais ils sont aussi très utiles pour les chercheurs qui étudient le fonctionnement du routage.

Le principe de base de NRTM ? Un IRR publie en HTTP ses données complètes, ainsi que des changements d'une version à l'autre (ce qui évite de télécharger la base entière à chaque fois). Un nouveau client va télécharger la base, le "*snapshot*", puis la tenir à jour avec les changement successifs. NRTM a donc des points communs avec le protocole Delta du RFC 8182. NRTM ajoute à cela des mécanismes de vérification d'intégrité et de passage à l'échelle.

Un serveur NRTM fournit un fichier "*Update Notification*" qui indique le numéro de la version actuelle de la base, ainsi que les endroits où récupérer "*snapshots*" et changements ("*Delta Files*"). Il y a à chaque moment un "*Snapshot File*" (la totalité de la base) et zéro, un ou plus "*Delta Files*" (les changements par rapport à un "*snapshot*"). Le "*Update Notification File*" est du JSON, plus précisement au format JWS (RFC 7515 ; JWS fait partie de l'ensemble des normes cryptographiques de JSON, JOSE <https://www.bortzmeyer.org/jose.html>). Les "*Snapshot File*" et "*Delta Files*" sont également du JSON, en l'occurrence du "*JSON Text Sequences*" (RFC 7464). Tous peuvent être comprimés en gzip (RFC 1952).

Le client doit connaitre l'adresse de l'"*Update Notification File*", et la clé publique utilisée pour la signer. Il charge l'"*Update Notification File*" (le RFC suggère de le faire toutes les minutes, c'est un petit fichier), et, en le lisant, il trouve les éventuelles "*Delta Files*" à télécharger, avec leur condensat cryptographique. Contrairement à l'"*Update Notification File*", "*Snapshot File*" et "*Delta Files*" ne sont jamais modifiées et peuvent donc être mémorisées par des intermédiaires HTTP.

Le gérant du serveur NRTM doit, la première fois, générer une clé publique ("*JSON Web Key*", RFC 7517), dont il annoncera la partie publique au format "*Textual Encoding of Subject Public Key*", de la section 13 du RFC 7468. Il ne reste plus qu'à mettre cette clé et les autres informations utiles au client sur le site Web de l'IRR. (Bon, il faudra aussi gérer les clés : faire des sauvegardes, remplacer les clés de temps en temps, etc.)

Le serveur doit aussi générer un `session_id`. C'est un identificateur qui servira aux clients à détecter un changement de serveur, ou bien une réinitialisation complète du serveur. Il est au format UUID (section 5.4 du RFC 9562).

Enfin, le serveur initialise sa base, en partant du numéro de version 1. À partir de là, le "*snapshot*" portera des numéros de version croissants, et chaque "*Delta File*" un numéro de version qui indique à