RFC 6837 : NERD: A Not-so-novel EID to RLOC Database

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 24 janvier 2013

Date de publication du RFC : Janvier 2013

https://www.bortzmeyer.org/6837.html

Dès qu'on sépare, dans un protocole réseau, le localisateur de l'identificateur <https://www.bortzmeyer.org/separation-identificateur-localisateur.html>, on a créé la nécessité d'une base de données pour mettre les deux en correspondance : une fois qu'on a l'identificateur, il faut bien trouver le localisateur. Et cette base doit être accessible depuis le réseau. Le protocole LISP n'impose pas une base particulière : son architecture permet de tester plusieurs types de base et le système NERD ("*Not-so-novel EID to RLOC Database*") est une de ces bases.

En pratique, LISP, décrit dans le RFC 6830[1], a un mécanisme de correspondance identificateur-¿localisateur favori, ALT (RFC 6836). Mais la base de données des correspondances est accédée via une interface standard (RFC 6833) et, derrière cette interface, on peut placer d'autres systèmes. NERD est nettement moins avancé qu'ALT (pas de mise en œuvre existante, juste un projet individuel) mais est intéressant parce qu'il explore une solution très différente. Le principe de NERD est de stocker la totalité des correspondances (ce n'est qu'un gros fichier, après tout, et relativement statique), et de la distribuer via les mécanismes habituels de distribution de fichiers (typiquement HTTP).

Question état d'avancement de NERD, il est amusant de noter que NERD était en fait le **premier** mécanisme de correspondance identificateur-¿localisateur proposé pour LISP. Pour diverses raisons, notamment organisationnelles à l'IETF, le RFC n'est publié que maintenant.

L'auteur estime que la méthode proposée fonctionne jusqu'à environ cent millions d'entrées dans la base. Que devra contenir cette base ? La correspondance entre les identificateurs (EID pour "*endpoint identifiers*") et les localisateurs (RLOC pour "*routing locators*"). Qu'est ce que consulte un routeur LISP lorsqu'il doit transmettre un paquet ? Ce paquet est à destination de l'identificateur 2001:db8:1337::1:af ? Regardons dans la base des correspondances, chic, on trouve que le localisateur correspondant en 192.0.2.254.

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc6830.txt

Encapsulons le paquet et envoyons-le à `192.0.2.254` qui se débrouillera ensuite (et, au passage, vous avez vu que les EID et les RLOC ont la forme physique d'une adresse IP et qu'un EID IPv6 peut avoir un RLOC IPv4 et réciproquement).

La plupart des autres systèmes de base de données de correspondance sont « à la demande ». C'est lorsque le routeur doit transmettre un paquet qu'il va se mettre en chasse de la correspondance. L'inconvénient est que cela va prendre du temps et que le premier paquet d'une communication risque fort de ne pas patienter assez longtemps. La perte du premier paquet d'une session (typiquement un TCP SYN) est particulièrement gênante car l'application attendra longtemps. Avec NERD, au contraire, le routeur a en permanence toute l'information. (Ce point du « premier paquet perdu » est développé dans la section 7.2, qui explique en quoi NERD traite ce problème mieux que ALT.)

Pour concevoir NERD, l'auteur a supposé acquis les points suivants (section 1.1) :

— Les données changent peu et uniquement lors d'opérations explicites d'un humain (par exemple, un site acquiert un nouveau FAI et donc un nouveau moyen d'être joint), pas lors d'événements techniques comme la panne d'un routeur ou d'un lien.
— Les données sont globales (identiques pour tout l'Internet) contrairement aux métriques de certains protocoles de routage (le coût d'une route, par exemple, dépend de la position).
— La mobilité des machines ou des réseaux (qui nécessite des changements fréquents des correspondances) ne sera pas traitée par NERD.

Ensuite, NERD définit quatre choses :

— Un format pour la base de données,
— Un format pour la distribution des mises à jour,
— Un mécanisme de distribution, qui sera généralement HTTP (RFC 7230),
— Un mécanisme de "*bootstrapping*" (NERD a besoin du réseau mais le réseau a besoin de NERD...).

Pour représenter les changements, un mécanisme aussi simple que le format patch traditionnel aurait pu être utilisé. Mais NERD utilise un format spécifique à la base de données LISP.

La section 2 explique comment fonctionne NERD :

— Une Autorité (volontairement, le RFC ne spécifie pas qui serait l'Autorité : peut-être un RIR) génère la base et la signe avec la clé privée dont la partie publique est dans son certificat X.509. (L'annexe A décrit comment signer avec OpenSSL : `openssl smime -binary -sign -outform DER -signer yourcert.crt -inkey yourcert.key -in database-file -out signature`.)
— La base est ensuite envoyée à un groupe de serveurs HTTP que les routeurs LISP connaissent.
— Les routeurs récupèrent la base et vérifient la signature (avec OpenSSL, `openssl smime -binary -verify -inform DER -content database-file -out /dev/null -in signature`). À partir de là, le routeur LISP peut fonctionner, il connait les correspondances identificateur-¿localisateur.
— L'Autorité peut générer des changements, reflétant les modifications de la base.
— De temps en temps, les routeurs reviennent interroger les serveurs pour demander les changements. Ceux-ci sont également signés. Une fois vérifiés, le routeur « patche » sa base avec les changeents.

Ainsi, un routeur d'entrée d'un tunnel LISP n'a jamais à attendre une résolution d'EID en RLOC. Il a toujours la totalité des informations. Ce gain en temps se paie par une consommation plus élévée de mémoire.

La section 2.3 discute qui pourrait être Autorité. NERD est compatible avec plusieurs mécanismes comme une Autorité unique et centrale (l'IANA ?), comme une oligarchie d'Autorités (les RIR), ou comme un système très décentralisé avec des tas d'Autorités différentes en concurrence.

Quant au format de la base, il figure en section 3. Il est fort simple, un en-tête avec les métadonnées. Parmi elles, la signature cryptographique au vieux format PKCS#7 du RFC 2315, un numéro de version permettant de voir si la base est plus récente qu'une copie locale, etc. Le format de la signature pourrait

passer à CMS (RFC 5652) dans le futur. Le certificat doit contenir un DNS-ID (voir RFC 6125) qui sera le nom identifiant l'Autorité.

Et les données ? Une simple liste d'identificateurs (EID) suivi chacun d'une liste de ses localisateurs (RLOC), encodée dans un format binaire décrit en section 3.1. Les changements (section 3.2) suivent un format simple : si un EID a des RLOC, il est ajouté à la base (ou il remplace complètement les valeurs existantes, s'il y en avait : ajouter un seul RLOC à un EID nécessite donc de diffuser tous les RLOC), sinon (si sa liste des RLOC est vide), il est retiré de la base.

La section 4 spécifie les URL à utiliser. La récupération initiale de la base `nerd.arin.net`, à partir d'un serveur `www.example.com` se fait avec `http://www.example.com/eiddb/nerd.arin.net/current/entire` et la récupération d'un changement se fait avec `http://www.example.com/eiddb/nerd.arin.net/current/chang` où 1105500 est le numéro de version de l'actuelle copie locale du routeur. Si le serveur n'a pas les changements depuis 1105500 (par exemple parce que c'est une trop vieille version), il a le droit de faire une redirection HTTP vers une autre version. Le routeur aura donc peut-être plusieurs étapes pour se mettre à jour. Attention, `www.example.com` ne doit pas mener à un identificateur LISP, sinon on rencontrerait un amusant problème de circularité (voir aussi la section 8.1).

Est-ce réaliste en pratique ? La section 5 analyse quantitativement les « dépenses » liées à NERD. D'abord, la taille de la base. Elle dépend évidemment des hypothèses faites. Pour une hypothèse basse, 100 000 EID et une moyenne de 4 RLOC par EID, la base ferait dans les 9 méga-octets. Un bien petit fichier selon les critères d'aujourd'hui. Ceci dit, une hypothèse haute, avec 100 millions d'EID et 8 RLOC par EID donnerait une base de 17 giga-octets, nettement plus difficile à distribuer. Il est difficile de prévoir qu'elle est l'hypothèse réaliste. Par exemple, il y a aujourd'hui dans les 400 000 routes dans la table de routage globale, donc LISP, s'il est un succès, aura certainement plus de 400 000 EID. Le RFC compte sur un déploiement progressif, estimant que, même si l'hypothèse haute est atteinte, cela ne sera pas avant de nombreuses années, et les routeurs auront alors davantage de RAM.

Des calculs similaires donnent une idée du nombre de serveurs HTTP qui seront nécessaires pour distribuer une telle base, et ses changements, sans devoir attendre des heures.

Et pour jouer le rôle des Autorités ? Notez que ce RFC parle très peu de l'avitaillement de la base (enregistrement d'EID et changement de leurs RLOC). Voir les sections 2.3 et 5.4 qui donnent une idée de ce que serait le travail de ces Autorités et les caractéristiques (notamment de sécurité) qu'elles doivent avoir.

Les sections 6 et 7 sont consacrées aux alternatives non retenues. Pourquoi n'avoir pas utilisé XML pour le format ? (Principale raison : taille de la base, avec l'encodage habituel de XML.) Et pourquoi n'avoir pas spécifiéé d'autres mécanismes de distribution que ce système de miroirs HTTP ?

Un tel mécanisme aurait pu être celui de NNTP (RFC 3977). Il a largement montré ses capacités à distribuer rapidement une grande quantité de données et le RFC ne dit pas clairement pourquoi il n'a pas été retenu.

Plus éloigné du principe de NERD aurait été l'utilisation du DNS pour récupérer l'information de correspondance entre EID et RLOC. Les avantages du DNS sont que le partage administratif des responsabilités et l'infrastructure technique sont déjà en place. Il a déjà été envisagé de l'utiliser pour le routage (RFC 1383). Notre RFC suggère quelque chose du genre :

```
; Deux RLOC pour 10.0.128.0/23
$ORIGIN 0.10.nerd.arpa.
128   EID2RLOC   mask 23 priority 10 weight 5 172.16.5.60
      EID2RLOC   mask 23 priority 15 weight 5 192.168.1.5
```

Mais le DNS n'est pas temps réel et, bien que cela ne soit pas forcément une gêne pour l'application (qui attend le résultat de la résolution DNS), c'est plus embêtant lorsque cette attente est dans le système de routage, laissant l'application dans l'ignorance de ce qui se passe. Et puis l'utilisation du DNS pour LISP ferait peser une contrainte sur les serveurs de noms (ne pas les numéroter avec des EID, qui obligeraient à avoir LISP pour trouver la base de données LISP).

Voilà, si vous aimez les jolis transparents avec des images et des citations érudites, l'auteur du RFC a mis en ligne son court exposé sur NERD <http://dimacs.rutgers.edu/Workshops/SecureRouting/slides/nerd-dimacs.pdf>.

Il n'y a apparemment pas de mise en œuvre de NERD aujourd'hui, ni côté Autorités, ni dans les routeur (les serveurs, eux, seraient des serveurs HTTP standard). Disons que NERD est une solution de rechange si ALT ou les autres mécanismes de correspondance ne tiennent pas leurs promesses. Après tout, LISP est expérimental...