RFC 6835 : LISP Internet Groper (LIG)

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/6835.html

De même que dig est la référence, connue de tous, des outils de débogage du DNS, le nom de **lig** va peut-être devenir célèbre dans le monde des opérateurs réseaux. lig est également un outil de débogage : il sert à interroger la base de correspondance identificateur-¿localisateur du protocole de routage LISP (RFC 9300 [1], et sans lien avec le langage de programmation).

En quoi consiste LISP ? C'est une architecture de séparation de l'identificateur et du localisateur <https://www.bortzmeyer.org/separation-identificateur-localisateur.html> sur l'Internet (RFC 9300). Actuellement, l'adresse IP sert à deux choses : elle identifie une machine unique dans l'Internet (**identificateur**), et elle indique où envoyer les paquets à destination de cette machine (**localisateur**). LISP sépare ces deux fonctions. Et, comme toute architecture de séparation identificateur/localisateur, LISP nécessite une **correspondance** ("*mapping*") entre les deux. Si on n'a que l'identificateur (EID - "*Endpoint ID*" - dans la terminologie LISP), il faut trouver le localisateur (RLOC - "*Routing Locator*" - en termes LISP) pour pouvoir envoyer les paquets. C'est le rôle du système de correspondance. Et lig permet d'interroger cette base de correspondances, pour voir son contenu et déboguer ainsi des problèmes.

Il existe plusieurs systèmes possibles pour cette correspondance, comme les très expérimentaux CONS ou NERD. Mais le plus répandu est le système officiel, **ALT** (RFC 6836). Comme il existe un protocole standard de communication avec le système de correspondance, « LISP Map-Server » (dont l'interface est normalisés dans le RFC 9301), lig va pouvoir interroger tous les systèmes, ALT et les autres. Voici un exemple simple, où on demande le RLOC de l'EID `153.16.4.1` :

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc9300.txt`

```
% lig 153.16.4.1
Send map-request to eqx-ash-mr-ms.rloc.lisp4.net for 153.16.4.1 ...
Received map-reply from 216.129.110.58 with rtt 0.11500 secs

Mapping entry for EID '153.16.4.1':
153.16.4.0/24, via map-reply, record ttl: 1440, auth, not mobile
  Locator                                 State     Priority/Weight
  216.129.110.58                          up        1/100
```

Et on trouve que c'est `216.129.110.58`, qui va donc recevoir les paquets encapsulés à destination de `153.16.4.1`.

Pour suivre le RFC, ou même simplement cet article, il vaut mieux apprendre par cœur le vocabulaire, en section 2. Les termes que je trouve les plus importants :

— Map-Server : un élement de l'infrastructure réseau qui connait la correspondance entre identificateurs (EID) et localisateurs (RLOC). C'est typiquement une grosse machine qui sera dans le réseau de l'opérateur et utilisera les systèmes cités plus haut (comme ALT) pour publier son bout de la base des correspondances. Chaque Map-Server ne connait qu'une partie de la base, celle-ci étant répartie.
— Map-Resolver : c'est le composant qui interroge les Map-Servers. Il sera sans doute séparé de l'ITR (voir plus loin), son client. C'est ce Map-Resolver qu'interroge lig.
— RLOC : le "*Routing LOCator*" est l'adresse IP de l'ETR (voir plus loin). Une fois le RLOC connu, grâce au Map-Server, l'ITR peut lui envoyer les paquets LISP. (Le RLOC peut être vu comme le localisateur de la destination. LISP, contrairement à HIP <https://www.bortzmeyer.org/hip-resume.html>, n'est pas de bout en bout mais est une solution dans l'infrastructure, qui n'implique que les routeurs. Donc, le localisateur est attaché à un routeur, pas à la destination.)
— EID : le "*Endpoint Identifier*" identifie la destination. On l'a typiquement trouvé dans le DNS.
— "*EID-to-RLOC cache*" : les protocoles permettant de gérer la base des correspondances entre EID et RLOC sont typiquement assez complexes, et on ne souhaite pas les faire tourner sur un routeur, qui a d'autres responsabilités. Ils vont donc être exécutés sur le Map-Server, que le routeur va interroger via le Map-Resolver. Mais ce trafic d'interrogation va être intense, potentiellement une requête à chaque paquet que le routeur transmettra. Le routeur doit donc avoir un cache des correspondances les plus récentes. Contrairement à la base que stocke le Map-Server, ce cache est de petite taille, incomplet, et change très vite (les entrées y sont typiquement créées et détruites avec les flux réseau).
— ITR : l'"*Ingress Tunnel Router*" est le routeur d'entrée du tunnel LISP (rappelons que LISP fonctionne en encapsulant les paquets IP dans un tunnel qui va de l'ITR à l'ETR).
— ETR : l'"*Egress Tunnel Router*" est le routeur de sortie du tunnel LISP.
— xTR ; lorsqu'on souhaite parler ensemble des ITR et des ETR, on utilise souvent le terme de xTR. Par exemple, « Un déploiement possible de LISP est de mettre les xTR dans le routeur directement connecté au client. »

La section 3 donne le principe général de fonctionnement de lig et sa syntaxe. lig se comporte un peu comme un ITR qui a reçu un paquet IP dont la destination est un EID, et qui se demande à quel RLOC envoyer le paquet. lig prend donc en argument un EID, fabrique un paquet LISP de type Map-Request, l'envoie, et attend une réponse Map-Reply. Là, contrairement au routeur, il se contentera d'afficher cette réponse à l'utilisateur. Il indiquera également le RTT de la requête et l'état de l'ETR (LISP surveille en permanence l'état des tunnels, cf. RFC 9300, sections 10 et 11 « "*Reachability*" », pour éviter d'envoyer les paquets à un trou noir).

Outre l'EID, lig prend deux paramètres optionnels, l'EID source (au cas où la réponse en dépendrait) et l'adresse d'un Map-Resolver à interroger (dans le premier exemple, le résolveur était indiqué via la variable d'environnement `LISP_MAP_RESOLVER`).

On peut se servir de lig, comme dans l'exemple plus haut, pour savoir quel est le localisateur d'un EID donné, mais on peut aussi l'utiliser pour se "*liguer*" soi-même, en interrogeant les serveurs pour vérifier que le site où on est a bien enregistré son préfixe EID.

La section 4 présente les deux implémentations de lig actuelles. Celle présente dans les routeurs Cisco (dans des versions expérimentales du code, seulement) est en section 4.1. La syntaxe générale (où les crochets indiquent une mention facultative) et où les termes entre chevrons sont des variables) est `lig <destination-EID> [source <source-EID>] [to <Map-Resolver>]`. Un exemple d'utilisation, tiré du RFC, est :

```
router# lig abc.example.com
     Send map-request to 10.0.0.1 for 192.168.1.1 ...
     Received map-reply from 10.0.0.2 with rtt 0.081468 secs

     Map-cache entry for abc.example.com EID 192.168.1.1:
     192.168.1.0/24, uptime: 13:59:59, expires: 23:59:58,
                     via map-reply, auth
       Locator          Uptime    State  Priority/Weight  Packets In/Out
       10.0.0.2         13:59:59  up     1/100            0/14
```

Et si on veut tester son propre EID (se liguer soi-même), sans avoir à le taper, on peut remplacer le `<destination-EID>` par `self` (ou `self6` pour IPv6). Par exemple :

```
router# lig self
     Send loopback map-request to 10.0.0.1 for 192.168.2.0 ...
     Received map-reply from 10.0.0.3 with rtt 0.001592 secs

     Map-cache entry for EID 192.168.2.0:
     192.168.2.0/24, uptime: 00:00:02, expires: 23:59:57
                     via map-reply, self
       Locator       Uptime    State  Priority/Weight  Packets In/Out
       10.0.0.3      00:00:02  up     1/100            0/0
```

En section 4.2 figure la version en logiciel libre de lig, qui tourne notamment sur machines Unix et est disponible en `<https://github.com/davidmeyer/lig>`. Sa syntaxe est un peu différente, `lig [-m <Map-Resolver>] <destination-EID>`. Elle ne permet pas de sélectionner l'adresse source. Voici un exemple sur une machine Debian :

```
% lig -m l3-london-mr-ms.rloc.lisp4.net  153.16.10.254
Send map-request to l3-london-mr-ms.rloc.lisp4.net for 153.16.10.254 ...
Received map-reply from 173.36.254.163 with rtt 0.23900 secs

Mapping entry for EID '153.16.10.254':
153.16.10.0/24, via map-reply, record ttl: 1440, auth, not mobile
  Locator                                 State     Priority/Weight
  173.36.254.163                          up        1/100
```

On y voit que les paquets à destination de l'EID `153.16.10.254` doivent être encapsulés dans LISP par l'ITR et transmis à l'ETR en `173.36.254.163`. Si vous essayez avec une adresse qui n'est **pas** un EID, vous recevez une réponse négative ("*Negative cache entry*") :

```
% lig -m l3-london-mr-ms.rloc.lisp4.net  10.1.2.3
Send map-request to l3-london-mr-ms.rloc.lisp4.net for 10.1.2.3 ...
Received map-reply from 195.50.116.18 with rtt 0.01200 secs

Mapping entry for EID '10.1.2.3':
10.1.0.0/16, via map-reply, record ttl: 15, not auth, not mobile
  Negative cache entry, action: forward-native
```

Et si vous ne recevez rien :

```
% lig -m l3-london-mr-ms.rloc.lisp4.net  153.16.10.254
Send map-request to l3-london-mr-ms.rloc.lisp4.net for 153.16.10.254 ...
Send map-request to l3-london-mr-ms.rloc.lisp4.net for 153.16.10.254 ...
Send map-request to l3-london-mr-ms.rloc.lisp4.net for 153.16.10.254 ...
*** No map-reply received ***
```

Alors le plus probable est que vous êtes derrière un pare-feu fasciste qui bloque les réponses. Pensez à autoriser le port UDP 4342, par exemple sur Linux :

```
# iptables --insert INPUT --protocol UDP --sport 4342 -j ACCEPT
```

(ou peut-être, sur un pare-feu avec état, l'état `RELATED` mais je n'ai pas testé.) tcpdump ne connait pas encore LISP donc vous ne verrez que deux paquets non analysés (notez que la réponse est venue d'une autre adresse IP que celle interrogée...) :

```
14:56:11.873961 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto UDP (17), length 88)
    217.70.190.232.50793 > 206.223.132.89.4342: UDP, length 60
14:56:12.051870 IP (tos 0xc0, ttl 110, id 31203, offset 0, flags [none], proto UDP (17), length 80)
    173.36.254.162.4342 > 217.70.190.232.57375: UDP, length 52
```

On peut aussi utiliser un "*looking glass*" (cf. section 5) comme `<http://lispmon.net/lig.cgi>`, pour interroger sur un EID particulier. Si on veut une vision générale de l'état de LISP, on peut regarder `<http://www.lisp4.net/status>` (état des résolveurs) et `<http://www.lisp4.net/lisp-site>` (liste de sites LISP et de résolveurs).