

RFC 9301 : Locator/ID Separation Protocol (LISP) Control-Plane

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 23 août 2024

Date de publication du RFC : Octobre 2022

<https://www.bortzmeyer.org/9301.html>

Comme pour tous les protocoles de séparation de l'identificateur et du localisateur <<https://www.bortzmeyer.org/separation-identificateur-localisateur.html>>, le protocole LISP, normalisé dans le RFC 9300¹, doit faire face au problème de la **correspondance** ("*mapping*") entre les deux informations. Comment trouver un localisateur, en ne connaissant que l'identificateur? LISP n'a pas de solution unique et plusieurs protocoles de correspondance peuvent être utilisés. La stabilité du logiciel des routeurs imposait une **interface** stable avec le système de résolution des identificateurs en localisateurs. C'est ce que fournit notre RFC 9301, qui spécifie l'interface, vue du routeur, et qui ne devrait pas changer, même si de nouveaux systèmes de correspondance/résolution apparaissent. Ce RFC remplace le RFC 6833. L'interface change assez peu mais le texte est sérieusement réorganisé, et la spécification a désormais le statut de norme et plus simplement d'expérimentation.

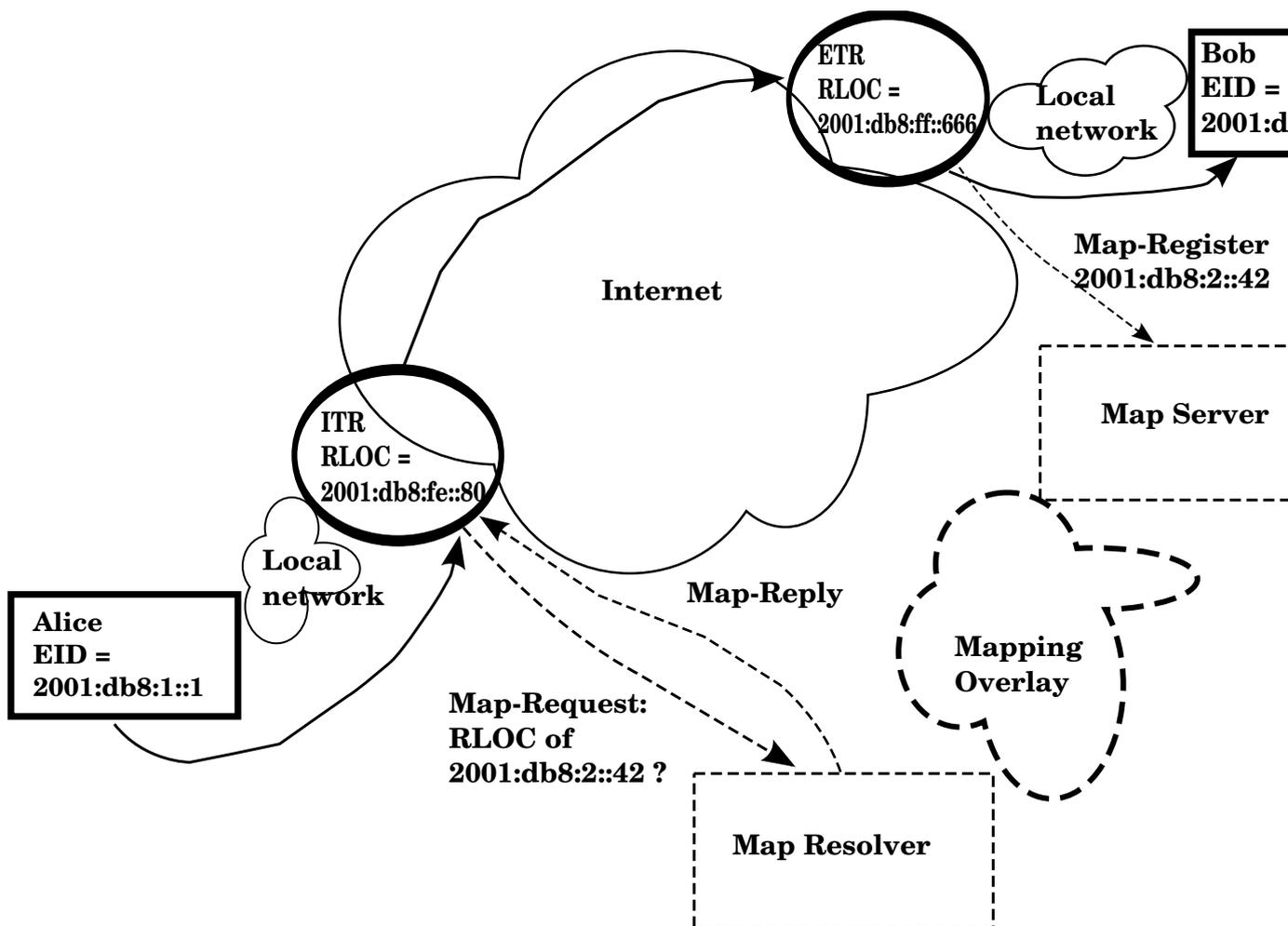
LISP prévoit deux sortes de machines impliquées dans la résolution d'identificateurs (les EID, "*Endpoint ID*") en localisateurs (les RLOC, "*Routing Locator*"). Ces deux machines sont les "*Map-Resolver*" et les "*Map-Server*". Pour ceux qui connaissent le DNS, on peut dire que le "*Map-Server*" est à peu près l'équivalent du serveur faisant autorité <<https://www.bortzmeyer.org/serveur-dns-faisant-autorite.html>> et le "*Map-Resolver*" joue quasiment le même rôle que celui du résolveur <<https://www.bortzmeyer.org/resolveur-dns.html>>. Toutefois, il ne faut pas pousser la comparaison trop loin, LISP a ses propres règles. Pour résumer en deux phrases, un routeur LISP d'entrée de tunnel (un ITR, "*Ingress Tunnel Router*"), ayant reçu un paquet à destination d'une machine dont il connaît l'identificateur (l'EID), va interroger un "*Map-Resolver*" pour connaître le localisateur (le RLOC, auquel l'ITR enverra le paquet). Pour accomplir sa tâche, le "*Map-Resolver*" fera suivre les requêtes au "*Map-Server*", qui la transmettra finalement au routeur de sortie du tunnel (l'ETR, "*Egress Tunnel Router*"), qui est la vraie source faisant autorité.

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

C'est entre le "*Map-Resolver*" et le "*Map-Server*" que se trouvent les détails du système de correspondance. Ils peuvent être connectés par ALT (RFC 6836), par CONS (RFC jamais publié <<https://datatracker.ietf.org/doc/draft-meyer-lisp-cons/>>), par NERD (RFC 6837), par DDT (RFC 8111) ou bien par tout autre système de résolution, existant ou encore à créer (ils ne peuvent pas être connectés avec simplement LISP, puisqu'on aurait alors un problème d'œuf et de poule, LISP ayant besoin de ALT qui aurait besoin de LISP...cf. section 8.1). Rappelez-vous que notre RFC 9301 ne décrit qu'une interface, celle des ITR et ETR avec les "*Map-Resolver*" et "*Map-Server*". Il est donc relativement court.

Comme avec toute technique nouvelle, il est prudent d'apprendre le vocabulaire (section 3, puis section 4 pour un survol général du système). Il y a deux types d'acteurs, les "*Map-Server*" et les "*Map-Resolver*" que nous avons déjà vu, et trois types importants de messages <<https://www.iana.org/assignments/lisp-parameters/lisp-parameters.xml#lisp-packet-types>>, *Map-Register* (un ETR l'envoie au "*Map-Server*" pour indiquer les RLOC des EID dont il est responsable), *Map-Request* (un ITR l'envoie à un "*Map-Resolver*" pour obtenir les RLOC; le "*Map-Resolver*" fait suivre jusqu'au "*Map-Server*", puis à l'ETR) et enfin *Map-Reply*, la réponse au précédent. Notons que ces types de messages ont leur description complète (avec leur format) dans le RFC 9300. Notez aussi que "*Map-Resolver*" et "*Map-Server*" sont des fonctions, et que les deux peuvent être assurés par la même machine, qui serait à la fois "*Map-Resolver*" et "*Map-Server*" (dans le DNS, un tel mélange est déconseillé).

Schéma général du système de correspondance LISP :



La section 8 de notre RFC plonge dans les détails. Accrochez-vous. Voyons d'abord le premier routeur LISP que rencontrera le paquet. On le nomme ITR pour "*Ingress Tunnel Router*". Les routeurs précédents traitaient l'adresse de destination du paquet comme une adresse IP ordinaire. L'ITR, lui, va la traiter comme un identificateur (EID pour "*Endpoint IDentification*"). L'EID n'est pas routable sur l'Internet. Il faut donc encapsuler le paquet en LISP pour l'envoyer dans le tunnel. La nouvelle adresse IP de destination est le localisateur (RLOC pour "*Routing LOCator*"). Pour trouver le localisateur, l'ITR va demander à un ou plusieurs "*Map-Resolver*". Il a été configuré (typiquement, à la main) avec leurs adresses IP (qui doivent être des localisateurs, pour éviter un amusant problème d'œuf et de poule; notez que plusieurs "*Map-Resolver*" peuvent avoir la même adresse, grâce à l'"*anycast*"). L'ITR ne connaît que le protocole de résolution, envoi d'une `Map-Request` et récupération d'une `Map-Reply` (en termes DNS, l'ITR est un "*stub resolver*"). L'ITR ne connaît donc pas les protocoles utilisés en interne par le système de correspondance, il ne connaît pas ALT (ou ses concurrents). Cette communication avec le "*Map-Resolver*" peut être testée et déboguée avec l'outil `lig` (RFC 6835).

La réponse du "*Map-Resolver*" ne sera pas forcément positive. L'ITR recevra peut-être une "*negative*" `Map-Reply`, envoyée en réponse si un "*Map-Resolver*" ne trouve pas de localisateur pour l'identificateur qu'on lui a passé. Cela veut dire que le site final n'utilise pas LISP, et qu'il faut alors router le paquet avec les méthodes habituelles d'IP. (Il n'est évidemment pas prévu que tout l'Internet passe à LISP du jour au lendemain, le routeur LISP doit donc aussi pouvoir joindre les sites non-LISP.)

Si la réponse est positive, l'ITR peut alors encapsuler le paquet et le transmettre. Comment le "*Map-Resolver*" a-t-il trouvé la réponse qu'il a envoyé? Contrairement aux routeurs LISP comme l'ITR, le "*Map-Resolver*" et le "*Map-Server*" connaissent le système de correspondance utilisé (si c'est ALT, ils sont tous les deux routeurs ALT) et c'est celui-ci (non traité dans ce RFC) qu'ils utilisent pour savoir s'il y a une réponse et laquelle.

Et, à l'autre bout du tunnel, que s'était-il passé? Le routeur de fin de tunnel (l'ETR, pour "*Egress Tunnel Router*"), avait été configuré par un administrateur réseaux avec une liste d'EID dont il est responsable. Pour que le reste du monde connaisse ces EID, il les publie auprès d'un "*Map-Server*" en envoyant à ce dernier des messages `Map-Register`. Pour d'évidentes raisons de sécurité, ces messages doivent être authentifiés (champ `Authentication Data` du message `Map-Register`, avec clés gérées à la main pour l'instant, avec SHA-256 au minimum), alors que les `Map-Request` ne l'étaient pas (la base de données consultée par les routeurs LISP est publique, pas besoin d'authentification pour la lire, seulement pour y écrire). Ces `Map-Request` sont renvoyés périodiquement (le RFC suggère toutes les minutes) pour que le "*Map-Server*" sache si l'information est toujours à jour. Ainsi, si un ETR est éteint, l'information obsolète dans les "*Map-Server*" disparaîtra en trois minutes maximum (des messages peuvent être perdus, le RFC demande donc de patienter un peu en cas de non-réception). Cela veut aussi dire que LISP ne convient pas forcément tel quel pour les situations où on exige une mobilité très rapide.

Notez que je ne décris pas tous les détails (comme la possibilité pour un ETR de demander un accusé de réception au "*Map-Server*", chose que ce dernier ne fait pas par défaut), voyez le RFC si vous êtes curieux.

Arrivés là, nous avons un "*Map-Server*" qui connaît les EID que gère l'ETR. Désormais, si ce "*Map-Server*" reçoit une demande `Map-Request`, il peut la faire suivre à l'ETR (si vous connaissez le DNS, vous avez vu que le `Map-Register` n'est pas tout à fait l'équivalent des mises à jour dynamiques du RFC 2136 : avec ces dernières, le serveur de noms qui a reçu la mise à jour répondra ensuite lui-même aux requêtes). Le "*Map-Server*" ne sert donc que de relais, il ne modifie pas la requête `Map-Request`, il la transmet telle quelle à l'ETR. Le rôle des "*Map-Resolver*" et des "*Map-Server*" est donc simplement de trouver l'ETR responsable et de lui faire suivre (sans utiliser l'encapsulation LISP) les requêtes, pas de répondre à sa place. Cela se fera peut-être dans le futur lorsque des mécanismes de cache seront ajoutés.

Pour le moment, les *"Map-Resolver"* n'ont pas de cache, de mémoire (section 4), une grosse différence avec le DNS (section 1).

La section 9 fait le tour des questions de sécurité liées au service de résolution. Comme les requêtes sont faites avec le format de paquets de LISP, elles héritent des services de sécurité de LISP comme le *"nonce"* qui permet de limiter les risques d'usurpation ou comme la sécurité LISP du RFC 9303. Par contre, comme pour les protocoles utilisés dans l'Internet actuel, il n'y a pas de vraie protection contre les annonces faites à tort (un *"Map-Server"* qui annoncerait un EID qui n'est pas à lui). C'est un problème très proche de celui de la sécurité de BGP et qui utilisera peut-être le même genre de solutions.

Notez qu'en théorie, l'interface spécifiée dans ce RFC pourrait servir à d'autres protocoles que celui du RFC 9300, comme par exemple GRE (RFC 2890) ou VXLAN (RFC 7348). Mais, pour l'instant, ce n'est pas le cas.

Il y a apparemment trois mises en œuvre. Outre l'outil de débogage lig (RFC 6835), il y a celle de Cisco pour ses routeurs, mais je ne connais pas les autres, sans doute dans des Unix.

Et les changements depuis le précédent RFC? Ils sont résumés dans la section 11 :

- Ajout du type de message `Map-Notify-Ack`, un accusé de réception,
- Plein de bits supplémentaires dans les en-têtes des messages,
- Dans les actions qu'un routeur peut prendre lorsqu'un paquet arrive, ajout de possibilités de rejet du paquet (section 5.4).