RFC 9299 : An Architectural Introduction to the Locator/ID Separation Protocol (LISP)

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 3 novembre 2022

Date de publication du RFC : Octobre 2022

https://www.bortzmeyer.org/9299.html

Le protocole réseau LISP ("*Locator/ID Separation Protocol*", rien à voir avec le langage de programmation du même nom) est normalisé dans le RFC 6830[1] et plusieurs autres qui l'ont suivi. Ce RFC 6830 est un peu long à lire et ce nouveau RFC propose donc une vision de plus haut niveau, se focalisant sur l'architecture de LISP. Cela peut donc être un bon point de départ vers les RFC LISP.

L'idée de base de LISP est de séparer l'identificateur du localisateur <https://www.bortzmeyer.org/separation-identificateur-localisateur.html>, comme recommandé par le RFC 4984. Un **identificateur** désigne une machine, un **localisateur** sa position dans l'Internet. Aujourd'hui, les adresses IP servent pour les deux, ne satisfaisant parfaitement aucun des deux buts : les identificateurs devraient être stables (une machine qui change de réseau ne devrait pas en changer), les localisateurs devraient être efficaces et donc être liés à la topologie, et agrégeables.

LISP a donc deux classes : les identificateurs, ou EID ("*End-host IDentifier*") et les localisateurs, ou RLOC ("*Routing LOCators*"). Les deux ont la syntaxe des adresses IP (mais pas leur sémantique). Un système de correspondance permet de passer de l'un à l'autre (par exemple, je connais l'EID de mon correspondant, je cherche le RLOC pour lui envoyer un paquet). LISP est plutôt prévu pour être mis en œuvre dans les routeurs, séparant un cœur de l'Internet qui n'utilise que les RLOC, d'une périphérie qui utiliserait les EID (avec, entre les deux, les routeurs LISP qui feraient la liaison).

En résumé (accrochez-vous, c'est un peu compliqué) :
— LISP peut être vu comme un réseau virtuel ("*overlay*") au-dessus de l'Internet existant ("*underlay*"),

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

— Les RLOC n'ont de sens que dans le réseau sous-jacent, l'"*underlay*",
— Les EID n'ont de sens que dans le réseau virtuel "*overlay*",
— Entre les deux, le système de correspondance,
— Dans le réseau sous-jacent, les RLOC servent de localisateur et d'identificateur,
— Dans un site à la périphérie, les EID servent d'identificateur et de localisateur pour les autres machines du site.

C'est ce côté « solution dans les routeurs » et donc le fait que les machines terminales ne savent même pas qu'elles font du LISP, qui distingue LISP des autres solutions fondées sur la séparation de l'identificateur et du localisateur, comme ILNP (RFC 6740).

Au passage, pourquoi avoir développé LISP ? Quel était le problème à résoudre ? Outre la beauté conceptuelle de la chose (il n'est pas esthétique de mêler les fonctions d'identificateur et de localisateur), le principal problème à traiter était celui du passage à l'échelle du système de routage de l'Internet, décrit dans le RFC 4984. En gros, le nombre de routes distinctes augmente trop vite et menace la stabilité des routeurs de la DFZ. Le but était donc, par une nouvelle architecture, de rendre inutile certains choix qui augmentent la taille de la table de routage (comme la désagrégation des préfixes IP, afin de faire de l'ingénierie de trafic). Le RFC 7215 décrit comment LISP aide dans ce cas.

La section 7 du RFC décrit les différents scénarios d'usage de LISP :

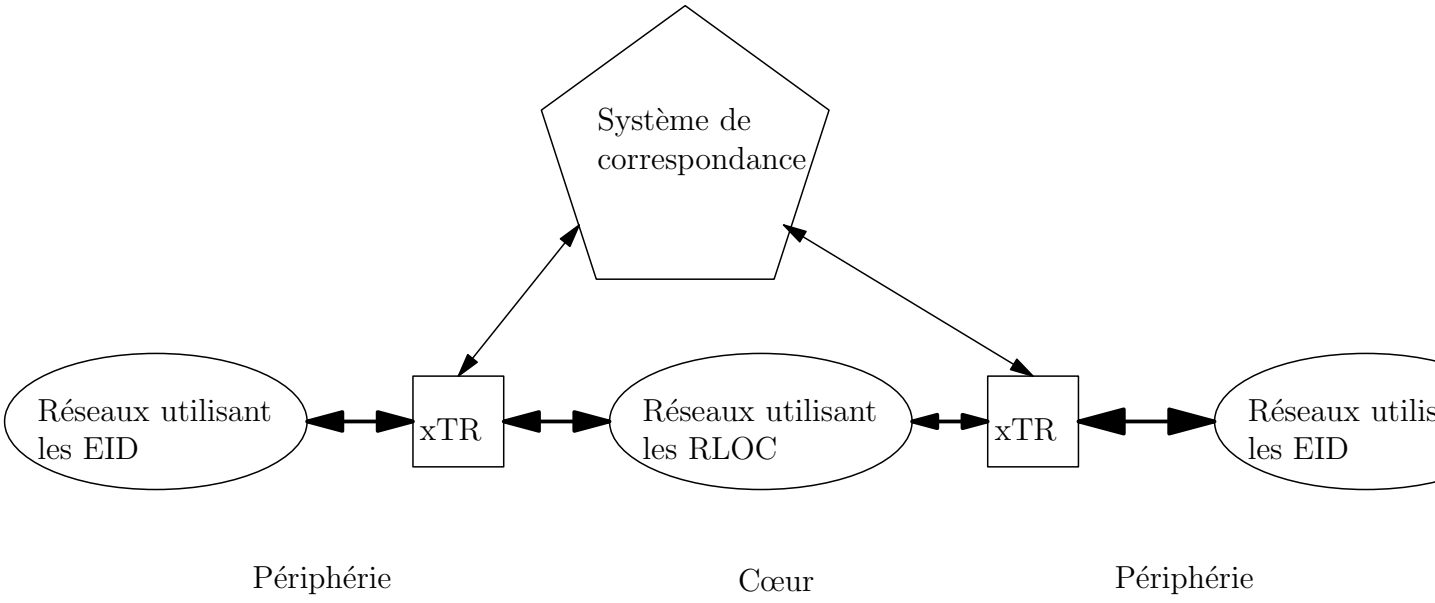
— Ingénierie de trafic : aujourd'hui, dans l'Internet classique BGP, quand on veut orienter le trafic entrant, le faire passer par un chemin donné, on utilise souvent la désagrégation des préfixes. Comme indiqué plus haut, cela aggrave la pression sur la table de routage globale. L'indirection que permet LISP dispense de cette solution : on peut publier dans le système de correspondance les RLOC qu'on veut.
— Transition vers IPv6 : les EID et les RLOC ont la forme syntaxique d'une adresse IP, v4 ou v6, et il n'y a pas d'obligation qu'EID et RLOC aient la même version. On peut avoir des EID IPv6 et des RLOC IPv4, et cela fournit un mécanisme de tunnel permettant de connecter deux sites LISP IPv6 au-dessus de réseaux qui seraient purement IPv4. L'avantage par rapport aux techniques de transition actuelles <https://www.bortzmeyer.org/transition-ipv6-guilde.html> est l'intégration dans une solution plus générale et plus « propre ».
— LISP permet également de faire des VPN, ou de gérer la mobilité de réseaux entiers (un réseau qui se déplace, et donc change de RLOC, c'est juste la correspondance dans le sous-système de contrôle qu'il faut changer, et tout les partenaires routeront vers les nouveaux RLOC).

La section 3 du RFC décrit en détail l'architecture de LISP (après, vous serez mûr·e·s pour lire les RFC LISP eux-mêmes, en commençant par le RFC 6830). Elle repose sur quatre principes :

— La séparation entre l'identificateur (le EID) et le localisateur (le RLOC),
— Deux sous-systèmes différents, le contrôle et les données, utilisant des protocoles différents, et pouvant évoluer séparement (enfin, dans une certaine mesure),
— Une architecture "*overlay*" : LISP est déployé sur un réseau virtuel au-dessus de l'Internet existant, ce qui évite les approches « table rase », qui ont une probabilité de déploiement à peu près nulle,
— Un protocole déployable de manière incrémentale, pas besoin d'attendre que tout le monde s'y mette, il y a des avantages à déployer LISP même pour les premiers à l'adopter (ce problème de déploiement est une des plaies de l'Internet actuel, comme on le voit avec IPv6).

La séparation entre identificateur et localisateur n'est pas faite au niveau de la machine individuelle, comme avec ILNP, mais à la frontière entre la périphérie de l'Internet ("*the edge*") et son cœur ("*the core*", en gros, la DFZ et quelques routeurs en plus). La périphérie travaille avec des EID (elle ne sait même pas que LISP est utilisé), le cœur avec des RLOC. Contrairement à ILNP, il n'y a donc pas une stricte séparation entre identificateurs et localisateurs : ils ont la même syntaxe (qui est celle d'une adresse IP, v4 ou v6) et, à part les routeurs d'entrée et de sortie des tunnels LISP, personne ne sait s'il utilise un EID ou un RLOC : les machines terminales manipulent des EID, les routeurs du cœur des RLOC, tout en croyant que ce sont des adresses IP ordinaires. Seuls les routeurs à la frontière entre les deux mondes connaissent LISP (et auront donc besoin d'un logiciel adapté).

Un Internet LISP est donc une série de « sites LISP » (des réseaux de périphérie accessibles par LISP) connectés par des tunnels entre eux, au-dessus du cœur actuel. Le routeur d'entrée du tunnel se nomme



ITR (pour "*Ingress Tunnel Router*") et celui de sortie ETR (pour "*Egress Tunnel Router*"). Le terme de xTR (pour « ITR ou bien ETR ») est parfois utilisé pour désigner un routeur LISP, qu'il soit d'entrée ou de sortie

Le sous-système des données ("*data plane*") se charge d'encapsuler et de décapsuler les paquets, puis de les transmettre au bon endroit. (Sa principale qualité est donc la rapidité : il ne faut pas faire attendre les paquets.) Les ITR encapsulent un paquet IP qui vient d'un site LISP (dans un paquet UDP à destination du port 4341), puis l'envoient vers l'ETR. À l'autre bout du tunnel, les ETR décapsulent le paquet. Dans le tunnel, les paquets ont donc un en-tête intérieur (un en-tête IP normal, contenant les EID source et destination), qui a été placé par la machine d'origine, et un en-tête extérieur (contenant le RLOC source, celui de l'ITR, et le RLOC de destination, celui de l'ETR puis, après l'en-tête UDP, l'en-tête spécifique de LISP). Rappelez-vous que les routeurs du cœur ne connaissent pas LISP, ils font suivre ce qui leur semble un paquet IP ordinaire. Les routeurs LISP utilisent des tables de correspondance entre EID et RLOC pour savoir à quel ETR envoyer un paquet.

Ces tables ont été apprises du sous-système de contrôle ("*control plane*", qui contient la fonction de correspondance - "*mapping*"), le routeur ayant un cache des correspondances les plus récentes. Cette fonction de correspondance, un des points les plus délicats de LISP (comme de tout système de séparation de l'identificateur et du localisateur) est décrite dans le RFC 6833. Son rôle peut être comparé à celui du DNS et de BGP dans l'Internet classique.

Une correspondance est une relation entre un préfixe d'identificateurs (rappelez-vous que les EID sont, syntaxiquement, des adresses IP ; on peut donc utiliser des préfixes CIDR) et un ensemble de localisateurs, les RLOC des différents routeurs possibles pour joindre le préfixe convoité.

Le RFC 6833 normalise une interface avec ce système de correspondance. Il y a deux sortes d'entités, le "*Map Server*", qui connait les correspondances pour certains préfixes (car les ETR lui ont raconté les préfixes qu'ils servent), et le "*Map Resolver*", qui fait les requêtes (il est typiquement dans l'ITR, ou proche). Quatre messages sont possibles (les messages de contrôle LISP sont encpasulés en UDP, et vers le port 4342) :

— `Map-Register` : un ETR informe son "*Map Server*" des préfixes EID qu'il sait joindre,
— `Map-Notify` : la réponse de l'ETR au message précédent,

— `Map-Request` : un ITR (ou bien un outil de débogage comme lig, cf. RFC 6835) cherche les RLOC correspondant à un EID,
— `Map-Reply` : un "*Map Server*" ou un ETR lui répond.

Un point important de LISP est qu'il peut y avoir plusieurs mécanismes de correspondance EID-¿RLOC, du moment qu'ils suivent les messages standard du RFC 6833. On pourra donc, dans le cadre de l'expérience LISP, changer de mécanisme pour voir, pour tester des compromis différents. Notre RFC rappele l'existence du système ALT (RFC 6836, fondé, comme BGP sur un graphe. Mais aussi celle d'un mécanisme utilisant une base « plate » (NERD, RFC 6837), un mécanisme arborescent nommé DDT (RFC 8111), des DHT, etc. On pourrait même, dans des déploiements privés et locaux, avoir une base centralisée avec un seul serveur.

ALT, normalisé dans le RFC 6836, est le système de correspondance « historique » de LISP, et il est souvent présenté comme le seul dans les vieux documents. Il repose sur BGP, protocole bien maitrisé par les administrateurs de routeurs, ceux qui auront à déployer LISP. L'idée de base est de connecter les serveurs ALT par BGP sur un réseau virtuel au-dessus de l'Internet.

DDT, dans le RFC 8111, lui, ressemble beaucoup plus au DNS, par sa structuration arborescente des données, et sa racine `<http://ddt-root.org/>`.

Évidemment, tout l'Internet ne va pas migrer vers LISP instantanément. C'est pour cela que notre RFC mentionne les problèmes de communication avec le reste du monde. Les EID ne sont typiquement pas annoncés dans la table de routage globale de l'Internet. Alors, comment un site pourra-t-il communiquer avec un site LISP ? Le mécanisme décrit dans le RFC 6832 utilise deux nouvelles sortes de routeurs : les PITR ("*Proxy Ingress Tunnel Router*") et les PETR ("*Proxy Egress Tunnel Router*"). Le PITR annonce les EID en BGP vers l'Internet, en les agrégeant le plus possible (l'un des buts de LISP étant justement d'éviter de charger la table de routage globale). Il recevra donc les paquets envoyés par les sites Internet classiques et les fera suivre par les procédures LISP normales. A priori, c'est tout : le site LISP peut toujours envoyer des paquets vers l'Internet classiques en ayant mis un EID en adresse IP source. Mais cela peut échouer pour diverse raisons (uRPF, par exemple) donc on ajoute le PETR : il recevra le paquet du site LISP et le transmettra.

Voici pour les principes de LISP. Mais, si vous travaillez au quotidien comme administrateur d'un réseau, vous avez sans doute à ce stade plein de questions concrètes et opérationnelles. C'est le moment de lire la section 4 de ce RFC. Par exemple, la gestion des caches : un routeur LISP ne peut pas faire appel au système de correspondance pour chaque paquet qu'il a à transmettre. Le sous-système des données tuerait complètement le sous-système de contrôle, si un routeur s'avisait de procéder ainsi. Il faut donc un cache, qui va stocker les réponses aux questions récentes. Qui dit cache dit problèmes de cohérence des données, puisque l'information a pu changer entre la requête, et l'utilisation d'une réponse mise en cache. Pour gérer cette cohérence, LISP dispose de divers mécanismes, notamment un TTL ("*Time To Live*") : l'ETR le définit, indiquant combien de temps les données peuvent être utilisées (c'est typiquement 24 h, aujourd'hui).

Autre problème pratique cruciale, la joignabilité des RLOC. C'est bien joli de savoir que telle machine a tel RLOC mais est-ce vrai ? Peut-on réellement lui parler ou bien tous les paquets vont-ils finir dans un trou noir ? Un premier mécanisme pour transporter l'information de joignabilité est les LSB ("*Locator Status Bits*"). Transportés dans les paquets LISP, ces bits indiquent si un RLOC donné est joignable par l'ETR qui a envoyé le paquet. Évidemment, eux aussi peuvent être faux, donc, s'il existe une communication bi-directionnelle, il est préférable d'utiliser le mécanisme des numniques `<https://www.bortzmeyer.org/nonce.html>`. Quand un ITR écrit à un ETR, il met un numnique dans le paquet, que l'ETR renverra dans son prochain paquet. Cette fois, plus de doute, l'ETR est bien joignable. Si l'ITR est impatient et veut une réponse tout de suite, il peut tester activement la joignabilité, en envoyant des `Map-Request`.

LISP est souvent présenté avec un modèle simplifié où chaque site est servi par un seul ETR, qui connait les EID du site et les annonce au "*Map Server*". Mais, dans la réalité, les sites sérieux ont plusieurs ETR, pour des raisons de résilience et de répartition de charge. Cela soulève le problème de leur synchronisation : ces ETR doivent avoir des configurations compatibles, pour annoncer les mêmes RLOC pour leurs EID. Pour l'instant, il n'existe pas de protocole pour cela, on compte sur une synchronisation manuelle par l'administrateur réseaux.

Enfin, comme LISP repose sur des tunnels, il fait face à la malédiction habituelle des tunnels, les problèmes de MTU. Du moment qu'on encapsule, on diminue la MTU (les octets de l'en-tête prennent de la place) et on peut donc avoir du mal à parler avec les sites qui ont une MTU plus grande, compte-tenu de la prévalence d'erreurs grossières de configuration, comme le filtrage d'ICMP. La section 4.4 de notre RFC décrit le traitement normal de la MTU dans LISP et ajoute que des mécanismes comme celui du RFC 4821 seront peut-être nécessaires.

Dans l'Internet d'aujourd'hui, une préoccupation essentielle est bien sûr la sécurité : d'innombrables menaces pèsent sur les réseaux (section 8 du RFC). Quelles sont les problèmes spécifiques de LISP en ce domaine ? Par exemple, certains systèmes de correspondance, comme DDT, sont de type "*pull*" : on n'a pas l'information à l'avance, on va la chercher quand on en a besoin. Cela veut dire qu'un paquet de données (sous-système des données) peut indirectement déclencher un événement dans le sous-système de contrôle (par la recherche d'une correspondance EID-¿RLOC afin de savoir où envoyer le paquet). Cela peut affecter la sécurité.

D'autant plus que le sous-système de contrôle sera typiquement mis en œuvre dans le processeur généraliste des routeurs, beaucoup moins rapide que les circuits électroniques spécialisés qui servent à la transmission des données. Un attaquant qui enverrait des tas de paquets vers des EID différents pourrait, à un coût très faible pour lui, déclencher plein de demandes à DDT, ralentissant ainsi sérieusement les routeurs LISP. Une mise en œuvre naïve de LISP où toute requête pour un EID absent du cache déclencherait systématiquement une `MAP-Request` serait très vulnérable. Une limitation du trafic est donc nécessaire.

En parlant du système de correspondance, il représente évidemment un talon d'Achille de LISP. Si son intégrité est compromise, si des fausses informations s'y retrouvent, les routeurs seront trahis et enverront les paquets au mauvais endroit. Et si le système de correspondance est lent ou en panne, par exemple suite à une attaque par déni de service, le routage ne se fera plus du tout (à part pour les EID encore dans les caches des routeurs). On peut donc comparer ce système de correspondance au DNS dans l'Internet classique, par son côté crucial pour la sécurité.

Il faut donc des « bons » "*Map Server*", qui suivent bien le RFC 6833 (notamment sa section 6) et, peut-être dans le futur, des "*Map Servers*" qui gèrent l'extension de sécurité LISP-Sec (si son RFC est publié un jour).

Dernier point de sécurité, le fait que LISP puisse faire du routage asymétrique (le chemin d'Alice à Bob n'est pas le même que celui de Bob à Alice). Rien d'extraordinaire à cela, c'est pareil pour lee routage Internet classique, mais il faut toujours se rappeler que cela a des conséquences de sécurité : par exemple, un pare-feu ne verra, dans certains cas, qu'une partie du trafic.

On trouvera plus de détails sur les attaques qui peuvent frapper LISP dans le RFC 7835.

Pour ceux qui sont curieux d'histoire des technologies, l'annexe A du RFC contient un résumé de LISP. Tout avait commencé à Amsterdam en octobre 2006, à l'atelier qui avait donné naissance au RFC 4984. Un groupe de participants s'était alors formé, avait échangé, et le premier "*Internet-Draft*" sur LISP

avait été publié en janvier 2007. En même temps, et dans l'esprit traditionnel de l'Internet ("*running code*"), la programmation avait commencé et les premiers routeurs ont commencé à gérer des paquets LISP en juin 2007.

Le groupe de travail IETF officiel a été ensuite créé <https://www.bortzmeyer.org/lisp-wg.html>, en mars 2009. Les premiers RFC sont enfin sortis en 2013.

LISP n'a pas toujours été comme aujourd'hui ; le protocole initial était plutôt une famille de protocoles, désignés par des numéros, chacun avec des variantes sur le concept de base. Cela permettait de satisfaire tous les goûts mais cela compliquait beaucoup le protocole. On avait LISP 1, où les EID étaient routables dans l'Internet normal (ce qui n'est plus le cas), LISP 1.5 où ce routage se faisait dans un réseau séparé, LISP 2 où les EID n'étaient plus routables, et où la correspondance EID-¿RLOC se faisait avec le DNS, et enfin LISP 3 où le système de correspondance était nouveau (il était prévu d'utiliser une DHT...). Le LISP final est proche de LISP 3.