RFC 6830 : Locator/ID Separation Protocol (LISP)

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/6830.html

Le protocole LISP (qui n'a rien à voir avec le langage de programmation du même nom), vise à résoudre un problème documenté dans le RFC 4984 [1] : les difficultés que rencontre le système de routage de l'Internet devant les pressions imposées par la croissance du réseau, et par le désir des utilisateurs de ne pas être lié à un unique fournisseur d'accès. Actuellement, tout changement de routage sur n'importe quel site se propage à tous les autres routeurs de la DFZ. Un tel système ne passe pas réellement à l'échelle (il ne peut pas croitre indéfiniment). LISP propose une solution (expérimentale, à l'époque), dans un ensemble de RFC dont ce RFC 6830 est le principal. Depuis, il a été remplacé par le RFC 9300.

Avant de plonger dans ce RFC, voyons les motivations pour LISP et ses principes essentiels (si vous préférez les lire dans le RFC et pas dans mon article, c'est en section 4). Aujourd'hui, les adresses IP ont deux rôles, localisation (où suis-je connecté au réseau) et identité (qui suis-je). Une adresse IP est un localisateur car changer de point d'attachement (par exemple parce qu'on change de FAI) vous fait changer d'adresse IP, et c'est un identificateur car c'est ce qui est utilisé par les protocoles de transport comme TCP pour identifier une session en cours : changer d'adresse IP casse les connexions existantes.

Le principal problème de cet approche concerne le routage. Un routage efficace nécessiterait une cohérence entre les adresses et la topologie du réseau, par exemple que deux sites proches sur le réseau aient des adresses proches. Mais on n'a pas cette cohérence actuellement. On notera qu'IPv6 ne résolvait pas ce problème, et délibérement (le cahier des charges d'IPv6 ne prévoyait pas de changer le modèle de routage).

Résultat, les routeurs doivent gérer bien plus de routes que nécessaire, augmentant leur prix (en attendant le jour où, même en payant, on n'arrivera pas à manipuler autant de routes, avec leurs changements fréquents). Le RFC 4984 insistait sur ce problème en affirmant que « *"The workshop participants*

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc4984.txt

believe that routing scalability is the most important problem facing the Internet today and must be solved, although the time frame in which these problems need solutions was not directly specified." »

Cette croissance de la table de routage peut être suivie sur le célèbre site de Geoff Huston <http://bgp.potaroo.net/>. Notez que la taille n'est pas le seul aspect, le rythme de changement (le nombre d'"*updates*" BGP par seconde) est au moins aussi important.

LISP vise donc à résoudre ce problème par une technique connue sous le nom de séparation du localisateur et de l'identificateur <https://www.bortzmeyer.org/separation-identificateur-localisateur.html> (son nom, « "*Locator/ID Separation Protocol*" », en dérive d'ailleurs, bien qu'il soit loin d'être le seul protocole dans cette catégorie). Au lieu que tous les préfixes IP aillent dans la DFZ, seuls les localisateurs, les RLOC ("*Routing LOCators*") y iront. Les identificateurs, les EID ("*Endpoint IDentifiers*") seront dans un nouveau système, le système de **correspondance** ("*mapping*"), qui permettra de trouver un RLOC à partir d'un EID. LISP est donc l'application d'un vieux principe de l'informatique : « Tout problème peut être résolu en ajoutant un niveau d'indirection. »

À quoi ressemblent RLOC et EID ? Physiquement, ce sont juste des adresses IP (v4 ou v6), avec une nouvelle sémantique.

Par rapport aux autres solutions de séparation de l'identificateur et du localisateur (la plus achevée étant HIP <https://www.bortzmeyer.org/hip-resume.html>), LISP s'identifie par les points suivants :

— Solution dans le réseau, pas dans les machines terminales. Seuls des routeurs (mais pas tous les routeurs de l'Internet) seront modifiés pour gérer LISP.
— Déployable de manière incrémentale (il n'est pas nécessaire que tout le monde passe à LISP).
— Pas de cryptographie (et donc pas plus de sécurité que l'IP d'aujourd'hui).
— Indépendant de la famille IP utilisée (v4 ou v6).
— Outre une solution au problème du passage à l'échelle du système de routage, LISP se veut aussi utilisable pour la mobilité et la virtualisation de réseaux (imaginez une machine virtuelle migrant d'un centre d'hébergement à un autre sans changer d'identificateur...).

Comment se passe l'envoi d'un paquet avec LISP ? Supposons qu'une machine veuille écrire à `www.example.com`. Elle utilise le DNS comme aujourd'hui, pour trouver que l'adresse est `2001:db8:110:2::e9` (c'est un EID, un identificateur, mais la machine n'a pas besoin de le savoir, les machines terminales ne connaissent pas LISP et ne savent même pas qu'elles l'utilisent). Elle envoie le paquet à son routeur habituel. À un moment dans le réseau, le paquet va arriver dans un routeur LISP (qui, lui, a été modifié pour gérer LISP). Il va alors chercher le RLOC (le localisateur). Il demande au système de correspondance (l'équivalent LISP du DNS) qui va lui dire « le RLOC de `2001:db8:110:2::e9` est `198.51.100.178` » (notez au passage que RLOC et EID peuvent être des adresses v4 ou v6). (L'information est stockée dans un cache du routeur, pour le prochain paquet.) Le paquet est alors encapsulé dans un paquet LISP qui est transmis en UDP (port 4341) à `198.51.100.178`. (En raison de ces deux étapes, correspondance puis encapsulation, LISP fait partie des protocoles nommés "*Map and Encap*".) `198.51.100.178` décapsule et le paquet suit ensuite un chemin normal vers la machine `2001:db8:110:2::e9`. Pendant le trajet dans le tunnel, le paquet aura donc deux en-têtes, l'**interne** (celui placé par la machine d'origine et qui utilise des EID dans les champs « Adresse IP source » et « Adresse IP destination ») et l'externe (celui placé par le routeur d'entrée du tunnel, et qui utilise des RLOC).

Si le système de correspondance avait répondu négativement « Ce n'est pas un EID, je n'ai pas de RLOC pour `2001:db8:110:2::e9` » ("*Negative map reply*") ? Cela aurait voulu dire que le site cible n'utilise pas LISP et, dans ce cas, on lui transmet le paquet par les mécanismes habituels d'IP.

Ainsi, pour prendre le scénario d'usage principal de LISP, un site qui veut être multi-homé n'aura pas besoin de BGP et d'annoncer ses routes dans la DFZ. Il aura ses identificateurs, ses EID, et les paquets

entrant ou sortant de son réseau seront encapsulés en LISP (le système de correspondance peut renvoyer plusieurs RLOC pour un EID, avec des préférences différentes, pour faire de l'ingénierie de trafic). Pour les routeurs de la DFZ, ce seront juste des paquets IP ordinaires. Seules les deux extrémités du tunnel sauront que LISP est utilisé.

Le système de correspondance de LISP n'est pas encore définitif : plusieurs choix sont possibles mais il existe un système privilégié, ALT (RFC 6836). Comme le DNS, il fonctionne en tirant les informations nécessaires (pas en les poussant vers tout le monde, comme le fait BGP), ce qui devrait lui donner une bonne capacité de croissance. De toute façon, LISP spécifie une **interface** vers le système de correspondance (RFC 6833) et les différents systèmes ont juste à mettre en œuvre cette interface pour qu'on puisse en changer facilement. Ainsi, ALT pourra être remplacé par un de ses concurrents, CONS, EMACS ou NERD (leurs noms sont des références au langage de programmation). NERD est documenté dans le RFC 6837.

LISP est aujourd'hui essentiellement promu <http://www.cisco.com/go/lisp> par Cisco, qui a monté un réseau mondial de test <http://www.lisp4.net>, qui compterait 140 sites. LISP est considéré comme expérimental, comme l'indique le statut de ce RFC (et la section 15 de notre RFC, qui liste les problèmes connus). Ses effets sur l'Internet ne sont pas encore complètement maîtrisés. LISP est à l'heure actuelle un protocole très controversé. (Ces avertissements ont été ajoutés au document peu de temps avant son adoption comme futur RFC.)

Ce RFC est un gros morceau (d'autant plus que d'autres parties de LISP sont dans d'autres RFC). Je ne vais pas le couvrir en entier. Mais quelques points méritent d'être gardés en tête :

— Un paquet dont l'adresse de destination est un EID ne peut être acheminé que via LISP. L'EID n'est pas routé sur l'Internet habituel. (Les EID peuvent être des adresses RFC 1918, par exemple.)
— Pour éviter que la base des EID ne pose les mêmes problèmes de croissance que la DFZ d'aujourd'hui, les EID seront agrégés, mais cela sera fait de manière indépendante de la topologie : si on change de FAI, cette agrégation ne changera pas.

Pour les fanas de format de paquets, la section 5 décrit l'encapsulation. LISP est indépendant de la famille d'adresses, donc on peut avoir un paquet IP où les EID sont IPv4 qui soit tunnelé avec des RLOC IPv6 ou bien le contraire. Devant le paquet normal, LISP va ajouter un en-tête IP standard pour les RLOC, où la source sera l'ITR (routeur d'entrée du tunnel) et la destination l'ETR (routeur de sortie du tunnel), puis un en-tête UDP (l'UDP a davantage de chances de passer les "*middleboxes*" que de l'IP mis directement dans l'IP), avec le port de destination à 4341, puis un en-tête spécifique à LISP et enfin le paquet original. Donc, pour résumer :

— En-tête du paquet original ("*inner header*" en terminologie LISP) : les adresses IP source et destination sont des identificateurs, des EID,
— En-tête vus par les routeurs situés entre l'ITR et l'ETR ("*outer header*") : les adresses IP source et destination sont des localisateurs, des RLOC.

L'en-tête spécifique à LISP contient notamment (section 5.3 si vous voulez tout connaître) :

— Cinq bits de contrôle, nommés N, L, E, V et I
— Si le bit N est à 1, un champ "*Nonce*" (section 6.3.1). Il s'agit d'un nombre tiré au hasard : si le destinataire d'un paquet peut le renvoyer, cela prouve qu'il avait reçu le message original (et qu'on parle donc bien au bon destinataire : ce numnique <https://www.bortzmeyer.org/nonce.html> sert à éviter les attaques en aveugle).
— Si le bit L est à 1, un champ "*Locator Status Bits*", qui indique l'état (joignable ou pas) des machines situées sur le site de départ du paquet.

Comme toutes les solutions à base de tunnel, LISP va souffrir de la mauvaise gestion de la PMTUD dans l'Internet d'aujourd'hui (cf. RFC 4459), l'en-tête LISP supplémentaire réduisant la MTU (cf. section 5.4 pour des solutions possibles).

La section 5 décrivait les paquets de données, ceux encapsulant les données envoyées par le site original. La section 6 couvre les paquets de contrôle, utilisés par LISP pour ses propres besoins, notamment le système de correspondance (cf. RFC 6833 pour les détails). On y retrouve l'utilisation d'UDP :

— "*Map Requests*", où le port de destination est 4342,
— "*Map Replies*",
— et quelques autres qui partagent des formats proches.

Il est évidemment essentiel qu'on sache si son correspondant est joignable ou pas. Comment cette « joignabilité » est-elle vérifiée ? La section 6.3 énumère les mécanismes disponibles. Entre autres :

— Les "*Locator Status Bits*" où un ITR (le routeur LISP à l'entrée du tunnel) indique si les sites qu'il contrôle sont joignables. Si on souhaite répondre à un message transmis en LISP, c'est une information cruciale.
— Les classiques messages ICMP comme "*Host Unreachable*". Comme ils ne sont pas authentifiés, les croire aveuglément est dangereux. Mais les ignorer totalement serait dommage.
— La réception récente d'un message "*Map Reply*" est une bonne indication que le site à l'autre bout fonctionne.

Mais on peut aussi tester explicitement, par le mécanisme "*Echo Nonce*" de la section 6.3.1. Le testeur émet un message LISP avec les bits N (numnique présent) et E (écho demandé), met le numnique à une valeur aléatoire (RFC 4086), et envoie le paquet. L'ETR à l'autre bout doit normalement renvoyer ce numnique dans son prochain paquet. Notons que cela teste la bidirectionnalité de la connexion. Si on n'obtient pas de réponse, cela peut signifier que la connexion est complètement coupée ou tout simplement qu'elle ne marche que dans un seul sens. Mais, surtout, l'absence de réponse peut indiquer le cas où l'ETR qui reçoit le trafic pour un site n'est pas le même routeur que l'ITR qui génère le trafic sortant. Dans ce cas, l'absence de réponse n'est pas un problème. Enfin, le routeur en face peut tout simplement être configuré pour ignorer les demandes d'écho.

Une autre solution pour tester est donc d'utiliser les messages du système de correspondance EID-¿RLOC, les "*Map Request*" et "*Map Reply*". Ces messages ont un bit P (pour "*probe*") qui indique que le demandeur est intéressé par la joignabilité du site demandé.

LISP impose donc des traitements supplémentaires, par rapport à ceux de l'IP classique. Est-ce que cela ne ralentit pas trop les routeurs ? La section 7 explore le problème et explique pourquoi LISP ne nécessite pas de changement du matériel de "*forwarding*" (les ASIC du routeur). La plupart des routeurs ont déjà du code prévu pour gérer des tunnels (encapsuler et décapsuler) rapidement.

Comment sera déployé LISP ? La section 8 décrit plusieurs scénarios possibles mais il faut regarder le futur RFC sur le déploiement de LISP pour avoir les détails. Principal problème : combien d'ITR et d'ETR pour un opérateur ? Grâce aux tunnels, on peut n'avoir qu'un seul ITR et un seul ETR pour tout le trafic. Cela poserait évidemment des problèmes de redondance et de performance. Mais avoir beaucoup de xTR peut poser des problèmes d'administration. Si les ITR sont largement automatiques (leur cache des correspondance EID-¿RLOC est bâti dynamiquement), avoir beaucoup d'ETR risque d'être compliqué à maintenir (car l'ETR doit avoir dans sa configuration une liste des EID qu'il va gérerà).

On peut aussi se demander si le premier routeur LISP utilisé par les clients doit être dans le réseau du FAI ou bien s'il ne devrait pas être dans le CPE (section 8.3).

Toujours dans la série des problèmes pratiques, que devient l'indispensable outil de débogage traceroute (section 9) ? Si les routeurs ne faisaient rien de spécial, un traceroute depuis un site LISP vers un autre montrerait le tunnel entre l'ITR et l'ETR comme un seul saut. Pour qu'on voit les routeurs dans le tunnel, il faut que LISP bricole un peu : les messages "*ICMP time exceeded*" générés dans le tunnel arriveront à l'ITR (c'est son adresse qui est la source des paquets encapsulés). L'ITR doit donc, pour faire des jolis traceroute, regarder le contenu du message ICMP, extraire le message original et donc l'adresse originale, puis envoyer un "*ICMP Time Exceeded*" à la vraie source.

Ça, c'était pour IPv6. En IPv4, c'est un peu plus compliqué car le paquet ICMP d'erreur ne contient pas la totalité du message-sonde de traceroute, même pas le maximum possible. L'ITR doit donc maintenir un cache des requêtes traceroute qu'il voit passer, indexé par un numéro qui sera utilisé comme port

source dans les requêtes encapsulées. Ainsi, lors de la réception du "*ICMP Time Exceeded*", l'ITR pourra retrouver la vraie source.

Pour ceux qui s'intéressent à la question de la mobilité, la section 10 y est consacrée. Un des avantages de la séparation de l'identificateur et du localisateur est justement de faciliter la mobilité. Un site ou une machine peut garder son EID en changeant de RLOC. Plusieurs cas :

— Si c'est tout le site qui bouge (un changement de FAI, par exemple, c'est une forme de mobilité, après tout), les machines du site ne voient rien, elles gardent leur EID et n'ont rien à faire. Il faut juste enregistrer la correspondance entre les EID et les nouveaux RLOC.
— Si c'est juste une machine qui bouge, on est dans le domaine de Mobile IP (RFC 6275 et RFC 4866). LISP peut aider mais ses interactions avec Mobile IP sont encore en cours d'exploration.

Un des gros problèmes que pose la séparation de l'identificateur et du localisateur est la sécurité : nouvelles attaques (par exemple contre le système de correspondance), nouveaux pièges (une machine qui utiliserait le vrai RLOC mais mentirait sur l'EID, par exemple), tout cela n'est pas encore bien connu et explique largement pourquoi les RFC sur LISP n'ont que le statut Expérimental. En attendant davantage d'expérience concrète , la section 12 examine les risques théoriquement présents dans LISP.

Comme avec toutes les techniques de tunnel, un émetteur peut facilement tricher sur l'adresse source interne (celle qui sera utilisée après décapsulation par l'ETR). Pour se protéger, un ITR devrait n'encapsuler un paquet que si l'adresse source est un EID qu'il contrôle. Et un ETR ne devrait transmettre un paquet que si la destination est un EID sous sa responsabilité.

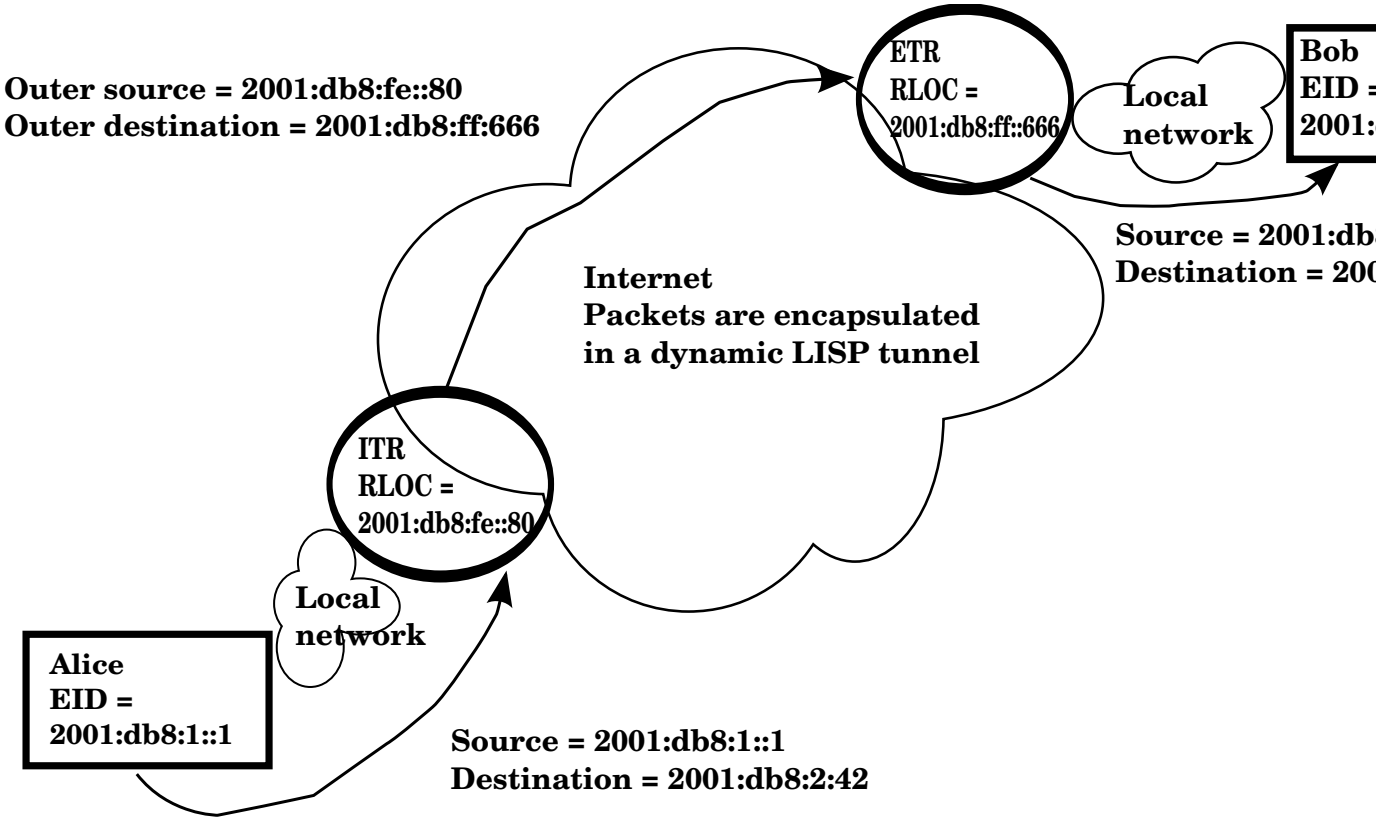
Le test de la réversibilité <`https://www.bortzmeyer.org/returnability.html`> (via les numniques, cf. section 3)) est essentiel contre ces risques. Sans ce test, un ETR pirate pourrait par exemple envoyer un "*Map Reply*" en réponse aveugle à un "*Map Request*", et le voir accepté, avec des données mensongères (naturellement, l'ITR n'accepte que des "*Map Replies*" qui sont en réponse à un "*Map Request*" de sa part). Avec ce système de numnique que le récepteur doit mettre dans sa réponse, un attaquant aveugle (qui n'est pas situé sur le chemin des paquets et ne peut donc pas les observer) aura donc peu de chances de réussir à faire accepter ses paquets.

En revanche, un attaquant situé sur le chemin, et qui peut observer les paquets qui passent, a la possibilité de commettre bien plus de dégâts. Mais c'est déjà le cas avec les protocoles actuels (par exemple, les numéros de séquence difficiles à deviner du RFC 6528 ne protègent pas TCP contre des attaquants situés sur le chemin).

Les attaques par déni de service sont évidemment possibles avec LISP : une des précautions recommandées est de limiter le trafic des "*Map Requests*" et "*Map Replies*". Autre attaque par déni de service, un ITR peut être victime d'une machine qui essaie de remplir la table des correspondances EID-¿RLOC du routeur. Il est donc important d'envisager ce cas, par exemple en permettant de garder dans le cache les entrées les plus fréquemment accédées (pour éviter qu'elles ne soient retirées automatiquement pour faire de la place). Mais il n'existe pas de solution miracle contre ce problème d'envahissement de cache.

On l'a dit, LISP a actuellement le statut Expérimental. La section 15 résume les problèmes connus à l'heure actuelle, et qui devront être traités pour obtenir un changement de statut :

— Recherche sur d'autres systèmes de correspondance EID-¿RLOC, au cas où ALT (RFC 6836) ne convienne pas.
— La gestion du cache des correspondances EID-¿RLOC, notamment en cas de fort usage (attaque par défi de service, par exemple) reste à explorer.
— Un RFC existe au sujet de l'interconnexion de LISP avec les sites non-LISP (RFC 6832) mais il n'y a guère d'expérience pratique.

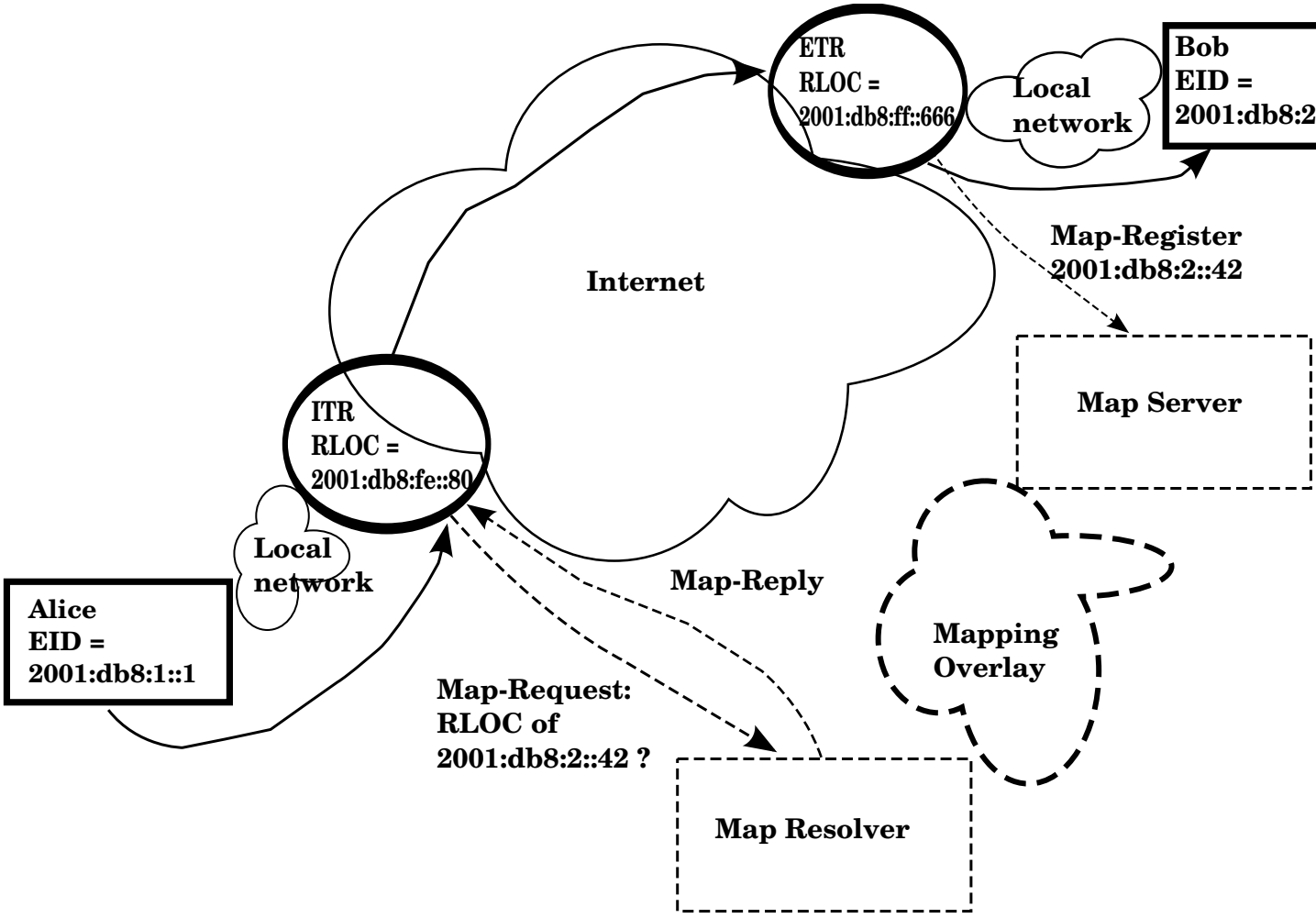


— Traditionnellement, les protocoles Internet séparaient strictement le contrôle et les données : un simple paquet IP ne pouvait pas modifier les routes, par exemple. Ce n'est plus tout à fait le cas avec LISP : un paquet de données arrivant à un ETR peut mettre à jour le cache de celui-ci (section 6, sur le "*gleaning*"). Il reste à voir quelles conséquences cela aura.
— L'Internet est un système très complexe, et les conséquences de LISP sur toutes les parties de ce système ne sont pas encore assez connues.
— Le RFC 6115 (section 2.2) avait formulé des critiques sur LISP, qui n'ont pas encore fait l'objet d'un traitement (montrer que la critique était injustifiée, ou bien modifier LISP pour en tenir compte).

Le fonctionnement de LISP est schématisé sur ce dessin : Alice a l'identificateur (EID) `2001:db8:1::1` et veut écrire à Bob qui a le `2001:db8:2::42` (dans la plupart des cas, Alice aura trouvé l'adresse de Bob dans le DNS, comme aujourd'hui). Ni Alice, ni Bob n'ont à connaître LISP, chacun croit utiliser de l'IP normal. Alice envoie donc un paquet standard, à destination de `2001:db8:2::42`. Mais, sur le trajet, il y a un ITR, un routeur LISP. Celui-ci va chercher (dans le système de correspondance, non montré ici) le RLOC (le localisateur) de Bob (ou, plus exactement, de son site). Une fois qu'il a trouvé `2001:db8:ff::666`, il encapsule le paquet derrière un en-tête LISP et envoie ce paquet à l'ETR, l'autre routeur LISP en face, `2001:db8:ff::666`. Les routeurs de l'Internet, entre l'ITR et l'ETR, ne connaissent pas LISP non plus et routent ce qui leur semble un paquet IP normal. Arrivé à l'ETR, le paquet est décapsulé et poursuit son chemin par du routage IP classique. Sur tout le schéma, seuls l'ITR et l'ETR sont des nouveautés LISP.

Modifions légèrement le schéma pour y mettre le système de correspondance : On y voir l'ITR demander à son résolveur « Quel est le localisateur de `2001:db8:2::42` ? » et son résolveur lui répondre. Le résolveur avait demandé au serveur qui avait reçu de l'ETR un enregistrement disans « Le localisateur de `2001:db8:2::42` est `2001:db8:ff::666` ». Entre le résolveur et le serveur se trouve le cœur du système de correspondance. LISP en a plusieurs possibles, comme le ALT du RFC 6836.

Où trouve-t-on du code LISP aujourd'hui ?



— Bien sûr dans IOS puisque LISP est l'enfant de Cisco. Voir le site Cisco <http://lisp.cisco.com/>.
— FreeBSD a OpenLISP <http://www.openlisp.org/>.
— Pour Linux, je ne connais que LispMob <http://lispmob.org/>, qui traite un besoin spécifique (la mobilité).
— Il existe un "*patch*" pour Wireshark : LISP dissector <http://www.cba.upc.edu/developed-tools/lisp-upc-tools/lisp-packet-dissector> (rien trouvé pour tcpdump, par contre).
— Il parait qu'un code Cisco mettant en œuvre LISP tourne sur Android.

Comme pour tous les protocoles fondés sur le principe de la séparation de l'identificateur et du localisateur, il est toujours utile, si on veut en comprendre les principes, de (re)lire l'article fondateur de Chiappa, « "*Endpoints and Endpoint names : A Proposed Enhancement to the Internet Architecture*" <http://www.chiappa.net/~jnc/tech/endpoints.txt> ». Autres articles à lire :

— Le nouveau RFC, qui remplace celui-ci, le RFC 9300.
— Le site Web officiel <http://www.lisp4.net/> des tests de déploiement de LISP. Plein d'informations et notamment une très bonne liste de sites LISP <http://www.lisp4.net/lisp-site/>.
— Un bon résumé de LISP <http://blog.ine.com/2010/07/05/a-high-level-overview-of-lisp/>.
— Le premier article <http://blog.fryguy.net/2011/04/07/lisp-locator-identifier-separation-p **pratique** sur le protocole. Avec instructions, commandes, et résultat (sur des routeurs Cisco). Parfait pour les ingénieurs qui ont du mal à se taper les RFC LISP mais veulent quand même comprendre comment ça marche. Très détaillé. Le même auteur a d'autres articles sur LISP comme

celui expliquant comment tunneler IPv6 sur IPv4 <http://blog.fryguy.net/2011/04/08/more-lisp-using-it-to-enable-ipv6-over-ipv4/>.

— Le compte-rendu <http://www.nanog.org/meetings/nanog50/presentations/Tuesday/NANOG50.Talk9.lee_nanog50_atlanta_oct2010_007_publish.pdf> de la connexion de Facebook à LISP (Facebook a été le premier gros site à sauter le pas).
— Les articles de Pattincon notamment ses Lisp Papers <http://blog.pattincon.com/lisp-papers/>.
— Le document de Cisco « *"Locator/ID Separation Protocol Overview - LISP - a new routing architecture"* <http://lisp.cisco.com/LISP_Overview.pdf> » est plutôt centré vers les décideurs, exposant les bienfaits de LISP, sans guère de détails techniques.
— Au contraire, « *"LISP - a next generation routing architecture"* <http://lisp.cisco.com/BRKRST-3045_Vegas2011.pdf> » est bien plus technique.