RFC 6836 : LISP Alternative Topology (LISP+ALT)

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/6836.html

Dans tous les systèmes de séparation de l'identificateur et du localisateur <https://www.bortzmeyer.org/separation-identificateur-localisateur.html>, le gros problème à résoudre est celui de la **correspondance** entre les deux. Sur le papier, cette séparation semble une idée simple et qui apporte des tas de bénéfices ("*multi-homing*" facile, plus besoin d'adresses PI, plus de rénumérotation des réseaux, etc), mais, dès qu'on rentre dans les détails, on se heurte au problème de la correspondance : comment trouver le localisateur lorsqu'on ne connait que l'identificateur, et de manière raisonnablement sécurisée ? Le protocole LISP a donc lui aussi ce problème et l'une des approches proposées (LISP est encore expérimental) est ALT ("*ALternative Topology*") dont le principe est de créer un réseau virtuel des routeurs LISP, avec le mécanisme de tunnels GRE, leur permettant d'échanger en BGP ces informations de correspondance entre identificateurs et localisateurs. Avec ALT, route et résolution de noms sont fusionnés. De même que BGP dans l'Internet habituel peut être vu comme une résolution préfixe-¿"*nexthop*", ALT est une résolution identificateur-¿localisateur faite en utilisant le routage.

Donc, le problème est le suivant : un ITR (premier routeur LISP qui traite un paquet) reçoit un paquet à destination d'un EID (un identificateur), il doit trouver l'ETR (dernier routeur LISP avant que le paquet ne repasse dans le routage habituel) à qui le passer et pour cela connaître son RLOC (le localisateur). Pour cela, l'ITR interroge un "*Map Resolver*" qui lui donnera la correspondance EID-¿RLOC. Mais comment le "*Map Resolver*" a-t-il trouvé l'information qu'il va servir ? Il existe plusieurs méthodes et notre RFC décrit l'une d'elles, ALT ("*ALternative Topology*").

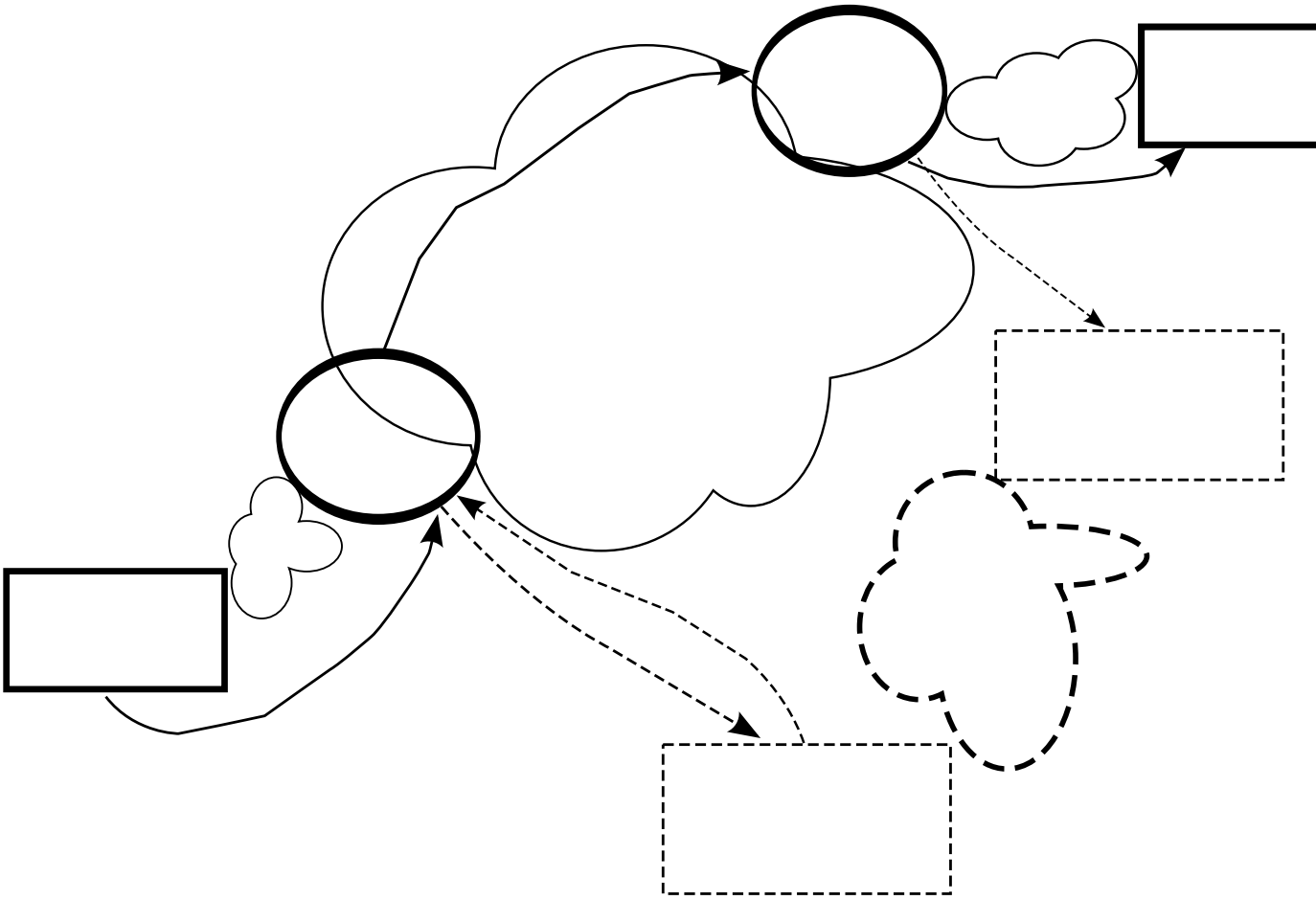
Pour éviter de réinventer la roue, ALT repose donc sur BGP (RFC 4271 [1]). Mais le protocole BGP n'est pas utilisé entre les routeurs habituels mais uniquement entre les routeurs ALT. Le réseau de ces routeurs est un réseau virtuel, construit avec des tunnels GRE (RFC 2784). Une fois le réseau virtuel, l'"*overlay*", construit, les routeurs ALT vont se communiquer en BGP la liste des préfixes EID, qui va

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc4271.txt

ainsi se propager partout. Les routes ainsi publiées mèneront à l'origine de l'annonce, un "*Map Server*" LISP qui connait l'ETR à qui demander la correspondance entre EID et RLOC. Notez donc bien qu'ALT ne distribue pas réellement les correspondances EID-¿RLOC mais les routes permettant de joindre celui qui connait la correspondance. Comme cela, il pourra recevoir une requête "*Map Request*" de LISP.

L'ITR typique, qui se retrouve avec un paquet IP destiné à un EID (l'identificateur) dont il ne connait pas le RLOC (le localisateur) ne fait en général pas partie de ALT. Il écrit à un "*Map Resolver*" (en utilisant l'interface standard décrite dans le RFC 6833) qui, lui, utilisera ALT (section 4.1). La réponse sera gardée dans un cache de l'ITR, pour accélerer les requêtes suivantes.

De même, ce n'est pas en général un ETR qui est membre de ALT pour diffuser les correspondances EID-¿RLOC mais un "*Map Server*", auprès duquel les ETR enregistreront leurs préfixes.



Comme tous les protocoles de correspondance (par exemple le DNS), ALT réussit donc à résoudre le problème de l'œuf et de la poule : pour trouver un RLOC, on a besoin d'écrire à la machine qui le connait mais, pour cela, il nous faut son RLOC... Dans ALT, le problème d'œuf et de poule est résolu en créant ce réseau virtuel, qui ne repose que sur les EID (les identificateurs). Ce réseau virtuel ne sert qu'à la résolution d'identificateur en localisateur, il ne transporte pas de données.

On l'a vu, ALT réutilise des techniques très classiques et bien connues, déjà mises en œuvre dans tous les routeurs (BGP et GRE). Cela ne veut pas dire que les routeurs ALT seront les mêmes routeurs que ceux qui transmettent actuellement le trafic Internet. Bien au contraire, le RFC recommande de dédier des routeurs à cette tâche ALT.

Comme tous les RFC sur LISP, ce RFC 6836 rappelle que le protocole est **expérimental**. On ne sait pas encore trop comment LISP, et ALT, réagissent à des problèmes comme :

— Le délai de résolution pour le premier paquet à destination d'un identificateur inconnu (les protocoles situés au dessus pourraient s'impatienter, ALT est largement piloté par la demande, le résolveur n'a donc aucune garantie d'avoir toutes les informations à l'avance).
— Pour éviter le problème ci-dessus, LISP permet de transporter des données utilisateur (celle du premier paquet de la session) dans la demande de résolution d'identificateur en localisateur. Est-ce que cela vaut la peine de compliquer ainsi le protocole ?
— Comment bâtir un graphe ALT ? Actuellement, la construction du graphe des routeurs BGP se fait à la main, via les achats de transit et les accords de "*peering*". Y a-t-il une meilleure méthode ?
— Et, naturellement, les risques de sécurité. Comment se passera la première attaque DoS contre LISP ?

Bref, il y a besoin de pratiquer pour répondre à ces questions, et c'est la raison des déploiements actuels de LISP.

Donc, après ces principes de base, place à la construction et au fonctionnement de ALT. D'abord, révisons le vocabulaire (section 2). Les termes les plus importants :

— ALT ("*Alternative Logical Topology*") : le réseau "*overlay*" bâti avec BGP et qui connecte tous les routeurs ALT. Étant un réseau virtuel, créé avec des tunnels, ses performances ne sont pas extraordinaires et il ne sert donc qu'à transmettre les "*Map Requests*", surtout pas à faire circuler le trafic normal.
— EID ("*Endpoint IDentifier*") : les identificateurs de LISP. Ils sont regroupés en préfixes par les routeurs ALT, pour limiter le nombre d'annonces à transporter (comme l'agrégation dans BGP, cf. RFC 4632).
— RLOC ("*Routing LOCator*") : les localisateurs de LISP. En pratique, des adresses IP.
— "*Map Server*" : un routeur ALT permettant l'enregistrement de préfixes EID par les ETR (routeurs non-ALT). C'est donc lui l'origine des EID dans ALT. Rappelez-vous que ALT fusionne résolution et routage. Un serveur d'identificateurs est un routeur. Le "*Map Server*" recevra des "*Map Requests*", qu'il transmettra à ces ETR.
— "*Map Resolver*" : un routeur ALT qui reçoit des demandes de résolution ("*Map Requests*") d'un ITR et les transmet dans le réseau ALT.
— ITR ("*Ingress Tunnel Router*") : un routeur LISP qui encapsule les paquets IP pour les transmettre à un ETR où ils seront décapsulés. Pour cela, il a besoin d'émettre des "*Map Requests*" vers un résolveur.
— ETR ("*Egress Tunnel Router*") : un routeur LISP qui décapsule les paquets LISP avant de les transmettre en IP classique. Il est également chargé de faire suivre les requêtes de résolution d'EID en RLOC ("*Map Requests*").

Comment tous ces jolis sigles sont-ils utilisés ? La section 3 décrit le modèle ALT. Il reprend les mécanismes de LISP, Deux types de paquets LISP peuvent entrer dans le réseau virtuel ALT : les "*Map Requests*", demande de résolution d'un EID en RLOC, et les "*Data Probes*", équivalents aux précédentes mais incluant en plus des données à faire suivre (le but est d'éviter à un routeur de garder en mémoire les données en attendant la réponse à la requête de résolution).

Comment est-ce que les informations rentrent dans le système ? La méthode de loin la plus courante (façon de parler puisque LISP est encore très peu déployé, disons la méthode dont on pense qu'elle sera la plus courante) est pour les ETR d'enregistrer les EID dont ils ont la charge auprès d'un "*Map Server*" (RFC 6833), qui les propagera alors dans ALT. Mais on verra peut-être aussi des routes statiques depuis les routeurs ALT vers les ETR ou bien des montages plus exotiques.

Et, pour envoyer un paquet à destination d'un EID, lorsqu'on est ITR ? La méthode « la plus courante » est que l'ITR écrive à son "*Map Resolver*", comme une machine Internet ordinaire écrit à son résolveur DNS pour trouver les adresses IP associées à un nom. Mais on aura peut-être la aussi des routes statiques (« si tu ne sais pas, envoie tous les paquets à tel ETR ») et d'autres configurations.

Donc, LISP n'impose pas un modèle unique mais recommande néanmoins fortement un modèle où les routeurs ALT sont au centre, communiquant entre eux, et où les ITR et ETR sont à l'extérieur de système de résolution, ne faisant pas d'ALT eux-mêmes mais transmettant les requêtes via les "*Map Resolver*" (questions des ITR) et les "*Map Server*" (réponses des ETR), dont les interfaces sont normalisées dans le RFC 6833. Un des avantages de ce modèle recommandé est qu'il permettra de tester d'autres systèmes que ALT et peut-être de le remplacer plus facilement (rappelez-vous que tout LISP est expérimental et que la partie « résolution d'identificateurs » est la plus expérimentale de toutes, donc très susceptible de changer).

Tout ce mécanisme ne gère pas le cas de la communication avec des sites non-LISP (tout l'Internet actuel) couvert dans le RFC 6832.

On l'a déjà dit, une des plus grosses discussions lors du développement de LISP avait été « le problème du premier paquet ». Lors d'une nouvelle connexion vers un identificateur, par exemple avec TCP, si le localisateur correspondant n'est pas dans les caches des routeurs, l'opération de résolution, a priori relativement lente, va obliger à garder le premier paquet dans les tampons de sortie du routeur, ou bien à jeter ce premier paquet. La première solution consomme de la mémoire et facilite les attaques par déni de service sur le routeur (un méchant pourrait commencer plein de connexions vers des EID injoignables, juste pour épuiser les ressources du routeur). La seconde est la seule réaliste (c'est ce qu'utilisent les routeurs du cœur de l'Internet pour les résolutions ARP, par exemple). Mais elle fait perdre un paquet crucial, celui d'ouverture de connexion. Cela obligera donc l'émetteur à attendre l'expiration d'un délai de garde, puis à réémettre, introduisant ainsi une latence considérable. C'est pourquoi des solutions où l'entiereté de la base des correspondances identificateur-¿localisateur serait gardée en mémoire de chaque routeur (ou, plus exactement, de chaque résolveur), avaient été proposées. ALT n'a pas suivi cette voie et propose donc autre chose : un type de paquets spécial, le "*Data Probe*" permet à la fois de demander une résolution (comme le fait un paquet "*Map Request*") **et** de transmettre le premier paquet. Cette solution a été très contestée (elle viole le principe « le réseau virtuel ne sert qu'à la résolution ») et l'une des choses qui sera suivie avec le plus d'attention dans le déploiement de ALT est l'utilisation des "*Data Probes*". La principale crainte est que ces "*Data Probes*", potentiellement bien plus gros que les "*Map Requests*" soient un bon vecteur de DoS pour le réseau virtuel ALT, qui n'est pas du tout conçu pour un tel trafic (cf. section 3.3). Le RFC recommande donc que les "*Data Probes*" ne soient pas activés par défaut et que le réseau limite vigoureusement leur débit.

La section 4 détaille l'architecture de ALT, présentée plus haut. Le RFC présente ALT comme "*push/pull*" mais je le décrirai personnellement comme étant plutôt "*pull*", comme le DNS, plutôt que "*push*" comme BGP. Pour tenir la charge dans un très grand réseau, une approche "*pull*" passe mieux à l'échelle. Dans ALT, la partie "*pull*" est la résolution elle-même ("*Map Request*" et réponse), la construction du réseau virtuel étant plutôt "*push*".

Les tunnels sont construits avec GRE (RFC 2784) mais ce n'est pas fondamental pour l'architecture d'ALT, qui pourrait utiliser d'autres types de tunnels dans le futur. (GRE est faible, question sécurité, par exemple.) L'avantage de GRE est qu'il est bien connu et bien maîtrisé. Deux routeurs ALT qui veulent communiquer doivent donc créer un tunnel entre eux (`ip tunnel add tun0 mode gre remote 10.2.3.4 local 172.16.0.76 ttl 255` sur Linux, `interface tunnel 0 \ tunnel source 10.2.3.4 \ tunnel destination 172.16.0.76` sur un Cisco, etc). Puis ils établissent une session BGP au dessus de ce tunnel. Ces sessions devront utiliser des nouveaux numéros de système autonome, pour éviter toute collision avec le graphe BGP actuel, celui des routeurs du cœur de l'Internet (section 6.1). Par contre, le RFC laisse ouverte la question de savoir s'il faut un nouveau SAFI ("*Sub-Address Family Identifier*", RFC 4760, qui sert aujourd'hui à distinguer les préfixes IPv4 de ceux IPv6). Cela serait souhaitable pour éviter qu'un préfixe EID soit injecté accidentellement dans la DFZ ou réciproquement. Mais cela compliquerait le déploiement, surtout avec les anciens routeurs qui ne connaitraient pas ce SAFI.

Comment sont alloués les identificateurs, les EID ? Attribués de manière hiérarchique, comme les adresses IP d'aujourd'hui, ils sont relativement statiques. On ne change pas d'EID quand on change d'opérateur réseau, par exemple. La topologie d'ALT, fondée sur les EID, n'a donc rien à voir avec celle de l'Internet, fondée sur les adresses IP avant LISP, et sur les localisateurs (RLOC) une fois LISP déployé. Et elle est bien plus stable. Des phénomènes ennuyeux du monde BGP comme le "*route flapping*" (un routeur qui annonce et retire frénétiquement un préfixe, faisant ainsi travailler tous les routeurs de la DFZ avant d'être amorti) seront donc probablement rares dans ALT (section 7.3).

Une note personnelle de gouvernance au passage : cela veut dire que la hiérarchie dans ALT ne suit pas des relations de "*business*" et il n'est donc pas sûr que l'agrégation de plusieurs préfixes EID dans un même "*Map Server*" se passe bien, puisque ces préfixes sont alloués à des organisations n'ayant aucun point commun (même pas le même opérateur). C'est clairement une des questions les plus ouvertes d'ALT, à l'heure actuelle (section 7.4). Si un routeur ALT a reçu de l'information sur les EID `10.1.0.0/24`, `10.1.64.0/24`, `10.1.128.0/24` et `10.1.192.0/24`, peut-il et doit-il agréger en un seul `10.1.0.0/16` ?

Les fanas de sécurité liront avec intérêt la section 10, qui lui est entièrement consacré. En gros, ALT a la même sécurité que BGP. Il a donc les mêmes vulnérabilités (un routeur ALT méchant peut annoncer n'importe quel préfixe EID, même s'il ne lui « appartient » pas) et on peut envisager les mêmes solutions (comme la protection des sessions BGP du RFC 5925, et pourquoi pas, demain, BGPsec).

Sinon, si vous êtes plus intéressé par les performances que par la sécurité, vous pouvez consulter un exposé sur l'évaluation des performances d'ALT `<http://www.ietf.org/proceedings/75/slides/lisp-4.pdf>`,

Il existe apparemment trois mises en œuvre différentes de ALT, dont une dans les routeurs Cisco.

Parmi les alternatives à ALT, il y a plusieurs propositions en cours de discussion à l'IETF, avec des jolis noms comme CONS, NERD ou DDT. Une solution évidente aurait été de choisir plutôt le système de correspondance le plus utilisé actuellement dans l'Internet, le DNS. Il existe des problèmes techniques avec le DNS mais la principale raison pour laquelle il n'a même pas été sérieusement considéré est que les gens qui travaillent quotidiennement sur des routeurs, et qui ont l'habitude de BGP et autres protocoles de routage, ne connaissent pas le DNS et ne l'aiment pas (ils l'assimilent souvent à un service de couche 7, loin de leurs problèmes opérationnels et envahis d'avocats). Comme tous les choix techniques, celui du système de correspondance a donc également une forte composante culturelle.