RFC 6741 : ILNP Engineering Considerations

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 10 novembre 2012

Date de publication du RFC : Novembre 2012

http://www.bortzmeyer.org/6741.html

Le RFC 6740 [1] décrivait l'architecture générale du mécanisme de séparation de l'identificateur et du localisateur <http://www.bortzmeyer.org/separation-identificateur-localisateur.html> ILNP. C'était le premier document à lire pour comprendre ILNP. Ce RFC 6741 est plus concret, exposant les questions d'ingéniérie autour d'ILNP. Il décrit surtout des points qui sont indépendants de la version d'IP utilisée, d'autres RFC couvrent les points spécifiques à IPv4 (RFC 6745 et RFC 6746) ou IPv6 (RFC 6743 et RFC 6744).

Donc, ILNP vise à fournir une meilleure architecture à l'Internet, en faisant évoluer IP vers une séparation de l'**identificateur** d'une machine (stable, indépendant de comment et par qui la machine est attachée au réseau) avec le **localisateur** (lié à l'attachement actuel au réseau, et pouvant donc changer). ILNP est décrit dans le RFC 6740 mais celui-ci ne donne qu'une vision de haut niveau, à laquelle manquent pas mal de détails. C'est le rôle de notre RFC que de combler ces manques. L'architecture ILNP peut avoir bien des incarnations concrètes, y compris par des approches « table rase » où on ne tient aucun compte de l'existant. Ce RFC 6741 présente une autre approche, plus progressive, où on essaie de déployer ILNP dans l'Internet actuel.

Vous avez bien lu le RFC 6740 d'architecture d'ILNP ou son résumé <http://www.bortzmeyer.org/6740.html> ? Alors, allons-y, plongeons nous dans les détails ILNP. D'abord, les **identificateurs** (section 2). Toute machine ILNP en a au moins un. Ils ne sont pas liés à une interface réseau et une machine avec N interfaces peut très bien n'avoir qu'un seul identificateur. (Notez toutefois qu'un identificateur peut être formé à partir d'une caractéristique d'une interface, comme l'adresse MAC. Mais ce n'est qu'un mécanisme de génération de l'identificateur, la carte réseau peut ensuite être changée sans affecter celui-ci.)

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc6740.txt

Les identificateurs sont des chaînes de 64 bits et la représentation la plus simple est donc un entier non signé de 64 bits (`uint64_t` dans un programme en C, par exemple). Comme les identificateurs d'interface d'IPv6 (section 2.5.1 du RFC 4291), ils utilisent la syntaxe EUI-64. Cette syntaxe décrit notamment deux bits, U et G, le premier indiquant si l'identificateur est globalement unique ou non, et le second indiquant s'il s'agit d'une adresse de groupe. Attention pour les programmeurs, rien n'empêche une machine de prendre l'identificateur qu'elle veut et, en pratique, des collisions peuvent se produire. À plusieurs reprises, notre RFC donne des conseils à ceux qui mettent en œuvre ILNP pour que ces collisions n'aient pas de conséquences graves.

Si le bit U est à 1, cela indique que l'identificateur dérive de l'adresse MAC, et est donc garanti unique via l'enregistrement à l'IEEE.

Si le bit U est à 0, cela veut dire que l'identificateur n'est pas dérivé d'un registre central et n'a donc qu'une signification locale. Cela permet à ILNP de se passer d'une dépendance vis-à-vis d'une autorité unique. Autre intérêt, on peut alors générer un identificateur qui soit une clé cryptographique, par exemple (RFC 3972). On peut aussi avoir des identificateurs temporaires, pour préserver la vie privée, comme on le fait en IPv6 avec les adresses temporaires du RFC 4941.

Normalement, ce RFC n'était censé couvrir que les points indépendants de la famille d'adresses utilisée. Néanmoins, on y trouve aussi l'encodage des localisateurs et identificateurs dans un paquet IP (section 3). En IPv6, dans les champs « adresse IP » (source et destination), le localisateur prend les 64 bits de plus fort poids, et l'identificateur le reste. Grâce à cet encodage, le routeur IPv6, qui ne tient compte que du préfixe, traitera le localisateur comme un préfixe IP normal et n'aura besoin d'aucune modification pour gérer de l'ILNP. Le format du paquet ILNPv6 est en effet exactement le même que celui du paquet IPv6 classique.

C'est plus complexe en IPv4, en raison de la petite taille des champs adresse. On met donc les localisateurs dans l'adresse (source et destination), afin que les routeurs IPv4 puissent travailler comme aujourd'hui, et l'identificateur (qui fait toujours 64 bits) doit être mis dans une option IP, comme décrit dans le RFC 6746.

Et au niveau 4 (transport), qu'est-ce qui doit changer ? La principale modification est que TCP ou UDP doivent lier l'état d'une session en cours uniquement à l'identificateur et non plus à une adresse. Ainsi, le tuple traditionnel de TCP {adresse IP locale, port local, adresse IP distante, port distant}, qui identifie une connexion en cours, doit être remplacé par {identificateur local, port local, identificateur distant, port distant}, afin de bénéficier d'ILNP (notamment de la possiblité de changer d'adresse sans casser la connexion).

Attention, TCP doit normalement remplir le champ « somme de contrôle » ("*Checksum*" dans l'entête TCP) en incluant les deux adresses IP, locale et distante. En ILNP, il ne doit plus utiliser que les identificateurs, les localisateurs étant remplacés, pour le calcul de la somme de contrôle, par une série de bits nuls.

Un aspect d'ingéniérie très important d'ILNP est l'**ILCC**, "*Identifier-Locator Communication Cache*". Cette structure de données, décrite en section 5, garde trace des détails permettant de maintenir une communication avec les CN ("*Correspondent Node*", les machines ILNP avec qui on parle). De tels caches existent déjà pour certains protocoles (par exemple pour ARP, sur une machine Unix, vous pouvez l'afficher avec la commande `arp -a -n`, pour NDP, sur Linux, avec `ip -6 neighbour show`). Et ceci sans parler des informations que doit garder TCP. Donc, le nouvel ILCC n'est pas forcément une grosse contrainte pour les machines.

Que stocke l'ILCC ?

— Le ou les Identificateurs du nœud local, en indiquant les préférences, et s'ils sont actuellement en service (utilisables pour une nouvelle session),
— Le ou les Localisateurs du nœud local, en indiquant les préférences, et s'ils sont actuellement en service,
— Le ou les Identificateurs de chaque correspondant (CN), en indiquant les préférences, leur validité, et s'ils sont actuellement en service,
— Le ou les Localisateurs de chaque correspondant, en indiquant les préférences, leur validité, et s'ils sont actuellement en service,
— Les numniques `<http://www.bortzmeyer.org/nonce.html>` à utiliser pour chaque correspondant, celui vers le CN et celui depuis le CN.

La validité peut être déterminée par divers moyens. Lorsque l'information a été apprise dans le DNS, c'est typiquement le TTL qui est utilisé. Toutefois, même lorsque le TTL expire, ce n'est pas une bonne idée de retirer les valeurs immédiatement (on risque de se couper d'un CN qui n'a pas mis à jour ses données assez vite). Il vaut donc mieux marquer un Localisateur comme « pas à jour mais peut-être encore utilisable si on n'a pas le choix ».

Et pour accéder à l'ILCC, quelle clé de recherche utilise-t-on ? L'identificateur, ce qui serait logique ? Raté. Pour les paquets qui contiennent un numnique, on utilise un tuple {Identificateur, Numnique} et pour les autres un tuple {Identificateur, Localisateur}. Le but de ce choix est de se prémunir contre les cas où deux machines utiliseraient le même identificateur (ce qui peut arriver, surtout avec le bit U à zéro).

On a vu qu'un des buts d'ILNP était de gérer proprement les cas où une machine change de connectivité (panne d'un lien, déplacement, etc). La section 6 explique les détails de ce cas. Lorsqu'une machine apprend qu'un de ses localisateurs ne marche plus, elle prévient ses CN ("*Correspondent Node*") par un message ICMP "*Locator Update*" (RFC 6743 et RFC 6745). Ce message permet de retirer ou d'ajouter des localisateurs. Il est authentifié par le numnique, qui doit être identique à celui d'une session en cours (sinon, le paquet entrant ne correspond à aucune entrée dans l'ILCC, voir la section précédente).

Ça, c'est pour les sessions en cours. Pour les nouvelles, si une machine veut être contactée, elle doit mettre à jour la liste de ses localisateurs dans le DNS. ILNP dépend donc assez fortement des mises à jour dynamiques du DNS (RFC 2136 et RFC 3007), ou d'un mécanisme similaire.

Le lecteur attentif a pu remarquer plus tôt que le localisateur joue le rôle d'un préfixe IP, le routage est fondé sur lui. Mais un préfixe IP n'a pas forcément une longueur fixe. En IPv6, on peut parfaitement avoir un Tier 1 qui annonce un /32 en BGP, qui route ensuite dans son IGP des /48 et finalement un réseau local d'un client de l'opérateur qui route sur des /64. ILNP ne change rien à cela (section 7). Le localisateur comprend en fait deux parties, de longueur variable, un préfixe de localisateur et un sélecteur de réseau. Dans le cas de l'annonce BGP citée plus haut, les 64 bits du localisateur se partagent entre 32 bits de préfixe et 32 bits pour choisir un réseau.

On a vu qu'ILNP dépend énormément du DNS. Il est donc logique que ce RFC compte une une section 8 consacrée à ce service. D'abord, comme on l'a vu, pour des raisons de sécurité, ILNP recommande fortement de sécuriser les mises à jour du DNS (RFC 3007). Cette technique marche dans de nombreuses mises en œuvre du DNS (par exemple, BIND ou Active Directory l'ont depuis des années). Les clients capables d'utiliser TSIG ou une technique similaire pour mettre à jour un serveur DNS sont disponibles pour toutes les plate-formes (comme `nsupdate` sur Unix). Un exemple figure dans mon article sur le RFC 6742.

Pour accéder à l'information disponible, quatre nouveaux types d'enregistrement DNS sont créés par le RFC 6742 :
— `NID` pour stocker l'identificateur,

— L32 et L64 pour stocker des localisateurs de 32 (pour IPv4) et 64 bits,
— LP, pour ajouter un niveau d'indirection vers le localisateur, apportant ainsi d'avantage de souplesse pour les changements (qui peuvent être fréquents) des localisateurs, qui peuvent être délégués à une autre zone.

Quels TTL utiliser pour ces enregistrements ? Le choix est particulièrement important pour les localisateurs, qui peuvent parfois changer vite. Dans certains cas, un TTL de quelques secondes seulement peut donc être une valeur raisonnable pour les L32 et L64. Si cela vous semble peu, rappelez-vous que des études comme celle de Alex C. Snoeren, Hari Balakrishnan, & M. Frans Kaashoek, « "*Reconsidering Internet Mobility*" <http://nms.csail.mit.edu/papers/migrate-hotOS.html> » (dans "*Proceedings of 8th Workshop on Hot Topics in Operating Systems*" en 2002) ou celle de S. Bhatti & R. Atkinson, « "*Reducing DNS Caching*" <http://www.dcs.gla.ac.uk/conferences/gi2011/p803-bhatti.pdf> » (dans "*Proc. GI2011 - 14th IEEE Global Internet Symposium*" en 2011) ont montré qur le cache dans le DNS est surtout utile pour les enregistrements d'**infrastructure** (type NS et colles, c'est-à-dire adresses IP des serveurs situés dans la zone qu'ils servent), pas pour les enregistrements « terminaux » comme AAAA ou MX.

Les enregistrements LP et NID sont bien plus stables et peuvent, eux, avoir des TTL de plusieurs jours. Dans le cas d'identificateurs temporaires et/ou cryptographiquement signés, un TTL plus court est toutefois à envisager.

À noter que le RFC recommande que les machines ILNP publient dans le DNS, non seulement les NID et L64 spécifiques d'ILNP mais aussi des plus traditionnels AAAA pour les machines non-ILNP. Un serveur DNS qui connait ILNP peut également envoyer d'autorité les NID et L64 dans la section additionnelle de la réponse DNS. Les machines non-ILNP ignoreront ces enregistrements inconnus mais les autres pourront en tirer profit.

L'expérience (douloureuse) d'IPv6 a montré que la possibilité d'un déploiement progressif, n'exigeant pas de tout le monde qu'il migre, était un élément nécessaire de tout nouveau protocole réseau. La section 10 de notre RFC regarde donc de près si ILNP est déployable progressivement. D'abord, les paquets ILNP, que ce soit ILNPv4 ou ILNPv6 sont indistinguables, pour les routeurs, les pare-feux et autres équipements, de paquets IPv4 et IPv6 classiques. Ils devraient donc passer sans problème sur ces équipements, sans qu'on change leurs configurations, leurs protocoles de routage, etc. ILNP, comme HIP <http://www.bortzmeyer.org/hip-resume.html>, peut en effet être déployé uniquement sur les machines terminales, par une simple mise à jour logicielle. Pour la même raison, des machines ILNP peuvent coexister avec des machines IP Classic sur un même lien.

Je trouve que le RFC est trop optimiste : l'expérience montre que les options IP, nécessaires à ILNP, passent mal dans l'Internet IPv4 actuel <http://www.bortzmeyer.org/options-interdites.html> et n'ont guère été testées en IPv6 (le RFC ne mentionne que le cas des pare-feux qui bloqueraient explicitement les options et pas celui de routeurs bogués). De toute façon, une exception est à prévoir, en ILNPv4 uniquement, les routeurs du réseau local devront être mis à jour pour gérer une extension d'ARP. Autre exception, certains routeurs NAT, selon la façon dont ils sont réglés, peuvent casser ILNP en réécrivant ce qu'ils croient être une adresse et qui serait un bout de l'identificateur.

Pour les applications, c'est plus compliqué. Si l'application est « bien élevée » (n'utilise que des noms de domaine, ne manipule pas explicitement d'adresses IP), elle devrait fonctionner en ILNP sans problème. Mais bien des applications ne sont pas bien élevées (FTP, SNMP...)

La section 14, consacrée aux applications et API revient en détail sur ces questions. D'abord, une bonne partie des applications existantes, notamment parmi celles écrites en C, utilise l'API dite "*BSD Sockets*". Elle n'est pas forcément incompatible avec ILNP. Par exemple, les applications IPv6 peuvent mettre la concaténation du localisateur et de l'identificateur dans les champs de l'API prévus pour

mettre une adresse IP <http://www.bortzmeyer.org/ip-data-structures.html> et cela passera, puisque c'est justement l'encodage du couple {Identificateur, Localisateur} sur le câble. Néanmoins, cette API est de trop bas niveau, même sans tenir compte d'ILNP (par exemple, elle oblige à avoir un portage d'IPv4 en IPv6 alors que cette modification dans la couche 3 devrait être invisible des applications <http://www.bortzmeyer.org/network-high-level-programming.html>). Souhaitons donc (et pas seulement pour ILNP) que les programmeurs passent à des API plus abstraites.

Tout le monde ne programme pas en C. Pour les autres langages, il est courant d'utiliser des API de plus haut niveau que ces "*sockets*". Ainsi, en Java, les clients Web vont plutôt faire appel à la classe `java.net.URL`. Ainsi, complètement isolés des couches basses, ils passeront à ILNP sans s'en rendre compte. Même chose pour des langages comme Ruby ou Python.

Le piège classique des API réseau est l'utilisation de **références**, lorsqu'Alice dit à Bob « va voir Charlie ». Ces références cassent pas mal de protocoles. Une solution pour ILNP serait de faire des références avec des FQDN.

Et pour communiquer entre machines ILNP et non-ILNP ? Le RFC recommande que les machines ILNP gardent également l'IP Classic, approche connue sous le nom de "*dual stack*" (être capable de parler les deux protocoles). Une longue coexistence d'ILNP avec des machines IP est prévue. Pour la machine qui initie la connexion, si elle trouve un `NID` dans le DNS, elle tente ILNP (en mettant l'option "*Nonce*"/numnique), sinon elle reste en IP Classic. Si la machine distante ne gère pas ILNP (malgré ce que raconte le DNS), elle rejettera l'option Nonce <http://www.bortzmeyer.org/nonce.html> avec un message ICMP (l'option a un type qui indique que le paquet doit être rejeté par ceux qui ne comprennent pas l'option, cf. RFC 2460, sections 4.2 et 4.5), qui permettra à l'initiateur de voir qu'il s'est trompé et doit réessayer en IP traditionnel.

Pour la machine qui répond aux demandes de connexion, si le paquet initial contient une option Nonce, c'est que le demandeur gère ILNP. Sinon, on s'en tient à l'IP de nos grand-mères.

Question sécurité, la section 9 est consacrée à IPsec. Il n'est pas très différent avec ILNP de ce qu'il est avec IP Classic. Le principal changement est que les associations de sécurité se font avec l'identificateur, pas avec l'adresse (elles ne changent donc pas si une machine change de localisateur). Ainsi, AH ne valide que l'identificateur (le localisateur n'est pas protégé, ce qui permet à ILNP+AH de fonctionner même en présence de NAT).

Autres questions de sécurité en section 11 : les auteurs estiment qu'ILNP n'est pas moins sûr qu'IP. Il ne l'est pas non plus forcément davantage. Un exemple typique est qu'une machine peut mentir sur son identificateur, comme elle peut mentir sur son adresse IP (les solutions sont également les mêmes, IPsec + IKE ou bien les CGA du RFC 3972).

Le principal risque nouveau est celui de faux messages ICMP "*Locator Update*". Ils pourraient provoquer bien des dégâts, par exemple en convaincant un CN d'envoyer tous les paquets d'une machine à un attaquant. C'est pour cela que l'option Nonce est absolument indispensable, même lorsque IPsec est utilisé. À noter que cette option ne protège **pas** contre un attaquant situé sur le chemin, puisque ce dernier peut alors écouter le réseau et lire le numnique (la situation est exactement la même en IP Classic, par exemple avec les numéros de séquence TCP, qui ne protègent que contre les attaquants hors du chemin). On est donc bien dans un cas « sécurité pas pire qu'avec IP Classic (mais pas meilleure non plus) ». Si on veut aller plus loin en sécurité, il faut IPsec ou un équivalent.

À noter aussi que les autres messages ICMP ("*Packet too big*", "*Destination unreachable*", etc), ne sont **pas** protégés par le numnique. Ces messages sont en général émis par des équipements intermédiaires,

genre routeurs, qui ne connaissent normalement pas le numnique. Comme en IP Classic, ils peuvent donc parfaitement être usurpés (cf. RFC 5927).

La sécurité inclus aussi le respect de la vie privée (section 12). Notre RFC renvoie au RFC 6740 qui discutait cette question dans le cadre d'ILNP.

Enfin, la section 13 contient diverses notes sur des problèmes pratiques. Notamment :

— Les délais de mise à jour lorsqu'une machine ILNP se déplace trop vite, par rapport à la réjuvénation du DNS <http://www.bortzmeyer.org/dns-propagation.html>. Mobile IP avait déjà ce problème. La mobilité d'un piéton, ça va, celle d'un avion va poser des problèmes.
— L'absence de garantie qu'on peut réellement joindre une machine via le localisateur indiqué. En effet, ILNP ne teste pas cela. Par exemple, un enregistrement `L64` dans le DNS peut être erroné et rien ne l'indique au client ILNP. Comme en IP Classic, on n'est jamais sûr qu'un paquet arrivera et les logiciels doivent en tenir compte.

Comme il n'y a pas encore eu de déploiement, même limité, même de test, d'ILNP, nul doute que la liste des problèmes et questions s'allongera lorsqu'on passera à l'expérience sur le terrain.