RFC 6115 : Recommendation for a Routing Architecture

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 22 février 2011

Date de publication du RFC : Février 2011

https://www.bortzmeyer.org/6115.html

Le mécanisme de routage dans l'Internet est une invention géniale. Bien qu'ayant été décrié par tous les experts des télécoms, unanimes dans leur appréciation comme quoi cela ne pourrait jamais marcher, ce mécanisme a permis la croissance d'un réseau de quatre machines aux États-Unis à des débuts, vers des centaines de millions aujourd'hui, réparties partout dans le monde. Néanmoins, rien n'est éternel en technologie. Tous les systèmes sont un jour remplacés, ne serait-ce que parce que de nouveaux besoins émergent et que de nouveaux problèmes apparaissent. Le groupe RRG <http://www.irtf.org/charter?gtype=rg&group=rrg> ("*Routing Research Group*") de l'IRTF est donc chargé d'explorer, pour le moyen et long terme, les solutions qui pourraient remplacer le routage actuel, notamment pour améliorer son passage à l'échelle, le cas des réseaux "*multihomés*" et la possibilité de faire de l'ingénierie de trafic. Ambitieux défi puisqu'il faut, non seulement concevoir un système meilleur que le système qui a assuré un tel succès, mais aussi le déployer, dans un réseau qui se méfie des nouveautés.

On ne s'étonnera donc pas que le RRG a à moitié échoué. Aucune idée géniale, créant un consensus par sa beauté, n'a émergé. De nombreuses propositions, souvent astucieuses, ont été présentées et discutées mais aucune n'a fait l'objet d'un consensus. Les présidents du groupe ont fini par publier le RFC tel quel, avec les différentes contributions, les points sur lesquels un accord s'est fait, et leur recommandation personnelle, qui est de partir de la proposition ILNP. Celle-ci a été normalisée en novembre 2012 dans les RFC 6740 [1] et suivants.

D'abord, un peu de contexte. Les documents sur lesquels se base le travail du RRG ("*Routing Research Group*") n'ont pas tous été publiés en RFC encore. Parmi eux, le RFC 6227, cahier des charges du projet. Les autres sont seulement des "*Internet-Drafts*". Le principal est l'exposé du problème, draft-narten-radir-problem-
Ces deux documents ont donné naissance à de nombreuses propositions. Attention en les lisant : il n'y a pas de censure à l'IRTF et toutes les propositions sont étudiées et discutées. Beaucoup provenaient d'une

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

personne isolée et, s'il existe parfois des génies solitaires et incompris, les propositions individuelles sont plus souvent l'œuvre de gens assez déconnectés de la pratique. On trouvera donc de tout dans ce RFC, hommage à l'ouverture de l'IRTF, plus qu'à sa capacité de sélection...

Il y avait en effet pas mal de zozos au RRG, des fois même incapables de formater un message électronique correctement (erreurs dans les citations, par exemple). Plusieurs propositions n'ont ainsi pas encore (?) connu le moindre début de mise en œuvre.

À noter également que les bruyants partisans des solutions de type « table rase » comme John Day <https://www.bortzmeyer.org/table-rase-et-john-day.html> ou comme le projet Clean Slate <https://www.bortzmeyer.org/science-et-vie-table-rase.html> n'ont pas participé du tout à cet effort. À leur décharge, il faut dire que la solution devait être déployable incrémentalement, ce qui contredisait leur postulat de départ, qui est d'ignorer l'existant. Mais cette non-participation reflète aussi leur difficulté à travailler sur des questions concrètes. Tant qu'on se contente d'interviews au New York Times, refaire l'Internet en partant de zéro est facile. S'il faut écrire de vraies normes, compréhensibles, et se confronter au jugement de ses pairs, il y a moins d'enthousiasme.

Je ne vais pas détailler chacune des nombreuses propositions, plutôt me servir de chacune pour exposer un défi particulier auquel est confronté l'architecte du futur réseau. Par manque de temps, j'ignorerai même complètement certaines des propositions.

Pour résumer ces documents de base, à quoi servait le groupe RRG <http://www.irtf.org/charter?gtype=rg&group=rrg> ("*Routing Research Group*")? Il avait été établit pour trouver une nouvelle architecture de routage pour l'Internet, rien que ça. Il s'est réuni à chaque réunion IETF de 2007 à 2010, a beaucoup discuté via sa liste de diffusion <https://www.irtf.org/mailman/listinfo/rrg>, mais, comme indiqué, n'a pas atteint de consensus. Les présidents du groupe, Lixia Zhang et Tony Li, deux experts expérimentés et reconnus, ont donc fini par jeter l'éponge et décider de tout publier, en ajoutant leur recommandation personnelle en faveur d'ILNP. Chaque proposition d'architecture ou de protocole a bénéficié d'un résumé (en général par son auteur, et la plupart de ces résumés sont de lourdes publicités avec peu de détails pratiques), résumé qui inclus obligatoirement les avantages de la proposition, mais aussi de ses coûts (il n'y a pas de déjeûner gratuit...), d'une critique par un autre membre du groupe et, parfois, d'une réponse de l'auteur à cette critique. (En pratique, je trouve que certaines critiques pinaillent sur quelques détails de la proposition, et n'analysent pas assez ses grandes idées et principes.)

Mais tout n'a pas été un flop : s'il n'y a pas eu d'accord sur une proposition concrète, le groupe a quand même mis en évidence plusieurs points importants, qui n'étaient pas évidents au début, mais qui ont rencontré un large accord. Ces points étaient (section 1.2) parfois de simple terminologie, parfois portant sur des concepts plus profonds. Par exemple :

terminologie Un **nœud** est soit un routeur, soit une machine terminale <https://www.bortzmeyer.org/terminal-host.html> ("*host*").

terminologie Un nom est une suite de bits utilisée pour désigner quelque chose (c'est vague mais c'est bien pour indiquer que le terme « nom » est très générique, et qu'il existe des termes mieux définis). Il existe des noms à des tas d'endroits de la pile des couches. Dans cette définition, le nom n'est donc pas forcément lisible et mémorisable par un humain.

— Une **adresse** est un nom qui mêle identification et indication de l'emplacement dans le réseau (évidemment pas dans l'espace physique). Les actuelles adresses IP sont dans cette catégorie.

— Un **localisateur** est un nom qui ne sert pas à l'identification mais uniquement pour exprimer une position dans le réseau (sans être une route complète vers l'objet). Actuellement, il n'existe pas de vrais localisateurs dans l'architecture Internet.

— Un **identificateur** est un nom qui ne contient pas d'information topologique (c'est-à-dire qu'il est complètement indépendant de la position du nœud, si on débranche une machine au Burkina Faso et qu'on la rebranche au Canada, ou si on passe de la connexion filaire avec le réseau local, à une connexion 3G, l'identificateur ne doit pas changer). Aujourd'hui, certains FQDN peuvent être considérés comme des identificateurs, selon cette définition. Même chose pour les clés SSH ou HIP <https://www.bortzmeyer.org/hip-resume.html>.
— L'intérêt de séparer les définitions des identificateurs et des localisateurs est qu'un des points consensuels du groupe RRG a été que la séparation de l'identificateur et du localisateur <https://www.bortzmeyer.org/separation-identificateur-localisateur.html> était à la fois faisable et souhaitable.
— Un autre consensus s'est fait sur l'importance du "*multihoming*", et qu'il soit possible à un coût raisonnable (la solution actuelle, avec adresses PI et BGP, ne passe pas tellement à l'échelle).

À propos de terminologie, je recommande au lecteur du RFC qui n'aurait pas suivi tous les travaux du RRG de bien lire la section 1.3, qui liste les sigles les plus courants. Tout le reste du RFC en fait une forte consommation et certains ne sont pas forcément familiers à l'ingénieur réseaux (comme ITR, "*Ingress Tunnel Router*", qui désigne le point d'entrée d'un tunnel, pour les solutions à base de tunnel ou EID, "*Endpoint IDentifier*", qui désigne l'identificateur de la machine de destination).

Maintenant, assez de préliminaires, place aux propositions concrètes. La première exposée est LISP, qui n'a rien à voir avec le langage de programmation du même nom et qui occupe la section 2. À noter que les « LISPiens » ont peu participé au RRG, ayant un groupe de travail IETF à eux <http://tools.ietf.org/wg/lisp> et leur proposition étant de loin la plus avancée, puisque implémentée et déployée. Par contre, aucun RFC LISP n'a encore été publié.

LISP fait partie des propositions dites **CES** pour "*Core-Edge Separation*", où une coupure très nette est faite entre le cœur de l'Internet (en gros, le monde de ceux qui ont aujourd'hui un routeur dans la DFZ) et les bords. Si LISP est largement déployé, seul le cœur aura encore des adresses IP globalement uniques et les bords auront des identificateurs uniques, qui ne seront pas des localisateurs. Un mécanisme d'annuaire ("*mapping*") permettra de trouver l'adresse d'un routeur capable d'acheminer les paquets pour un identificateur donné et ce routeur sera le point d'entrée d'un système de tunnels, qui permettra de router ce paquet. Le tunnel n'est pas forcément fourni par le FAI, et on pourrait imaginer un nouveau métier dans l'Internet, celui de fournisseur de tunnels. Comme avec toutes les autres propositions de séparation identificateur-localisateur, les localisateurs, plutôt dépendants du FAI, sont fortement agrégeables, les identificateurs sont stables (on ne les change pas en changeant de FAI, ce que les LISPiens appellent "*PI for all*"). LISP ne nécessite aucune modification des machines terminales (c'est une solution purement dans les routeurs et encore, pas dans tous, uniquement les routeurs d'entrée et de sortie de tunnel). Ce type de solution est aussi appelée "*Map-and-Encap*" car on cherche d'abord une correspondance entre identificateur et localisateur ("*Map*") puis on encapsule le paquet ("*Encap*") pour la traversée du tunnel.

Notez que cette CES, cette séparation entre cœur et bords changerait le modèle classique de l'Internet, fondé sur le même protocole IP partout, du réseau local de M. Michu à celui du plus gros Tier-1. Cela rappelerait les réseaux des PTT où X.25 ne servait qu'à l'accès, d'autres protocoles (comme X.75) étant utilisés à l'intérieur du réseau de l'opérateur.

Et, comme toutes les solutions fondées sur la séparation entre identificateur et localisateur <https://www.bortzmeyer.org/separation-identificateur-localisateur.html>, LISP nécessite un système de correspondance ("*mapping*") entre les deux, permettant de trouver, lorsqu'on cherche à joindre la machine d'identificateur X, à quel routeur Y il faut envoyer le paquet. Pour des raisons en partie idéologiques, LISP n'a pas considéré le DNS pour cette tâche et a plusieurs systèmes de correspondance possibles, le plus avancé se nommant ALT (pour "*ALternative Topology*"). La tâche serait relativement simple si le réseau ne changeait pas mais la réalité étant qu'il y a des pannes, des reconfigurations, etc, une bonne partie de la complexité de LISP et de sa fonction de correspondance va être

dans la détection et le traitement des pannes (pas question d'envoyer des paquets à un ITR en panne). Ce problème, là encore, est commun à toutes les propositions de séparation de localisateur et d'identificateur.

C'est donc logiquement sur ces fonctions (correspondance entre identificateur et localisateur, et détection puis contournement de pannes) que se focalisent les critiques. Alors que LISP ne change pas énormément le modèle d'IP, le déploiement d'un tout nouveau système de correspondance, non testé, et qui doit encaisser d'énormes quantités d'information changeant souvent, est vu comme le plus gros risque de LISP. Parmi les problèmes posés : que va t-il advenir du premier paquet, celui pour lequel le routeur ne connait pas encore de localisateur ? Le temps que le système de correspondance l'obtienne (et une requête dans ALT peut avoir un long chemin à parcourir), le paquet va devoir attendre. Les gros routeurs, ne souhaitant pas gérer de longues files d'attente, abandonneront probablement immédiatement de tels paquets, comme ils le font aujourd'hui lorsqu'il n'y a pas d'entrée ARP pour le destinataire. La source finira par réemettre et, cette fois, la table aura été remplie et le second paquet sera acheminé. Certains protocoles pourraient mal supporter cette « perte du premier paquet » (qui n'est pas non plus un problème spécifique à LISP). La réponse des LISPiens est que le problème n'est pas si grave que cela, les caches des routeurs se peupleront vite de l'essentiel des correspondances, ne serait-ce que grâce aux pirates "*scannant*" le réseau. Quant à ALT, il n'est qu'une des propositions, LISP offre le choix de plusieurs systèmes de correspondance identificateur -¿ localisateur comme TREE. Le modèle de LISP ne repose pas sur une fonction de correspondance unique.

Quant à la complexité de détection des pannes, nécessitant des échanges spécifiques entre les ITR et les ETR (les routeurs d'entrée et de sortie des tunnels), elle est vue comme inévitable, et, comme LISP fonctionne déjà sur le terrain, l'argument massue des LISPiens est que « ça marche ».

Le second système présenté, RANGI ("*Routing Architecture for the Next Generation Internet*", présenté par X. Xu), en section 3, est, au contraire de LISP, une solution basée sur les machines terminales et non plus sur les routeurs. Comme HIP <https://www.bortzmeyer.org/hip-resume.html>, RANGI introduit un identificateur propre à la machine et les connexions TCP sont liées à cet identificateur et plus aux adresses IP. Mais, alors que l'identificateur de HIP est « plat », sans structure, ce qui le rend difficilement résolvable en localisateur (et difficilement filtrable par les pare-feux), celui de RANGI est hiérarchique, pour pouvoir être résolu en localisateur par les méthodes arborescentes qui ont fait leur preuve, par exemple avec le DNS (alors que HIP doit utiliser des techniques moins éprouvées comme les DHT).

Comme HIP, RANGI ne connait que la distinction entre routeurs et machines terminales et n'a pas de distinction entre le cœur du réseau et ses bords. On parle donc de solution **CEE** ("*Core-Edge Elimination*"). Comme toute solution « côté machines terminales », RANGI nécessitera une mise à jour de tous les systèmes d'exploitation. La question de savoir s'il vaut mieux modifier toutes les machines terminales (plus nombreuses, pas forcément gérées, mais moins sensibles et plus souvent changées pour une neuve) ou tous les routeurs (moins nombreux, gérés par des professionnels mais souvent difficiles à remplacer ou à mettre à jour) a été souvent discutée dans le RRG mais sans résultat ferme.

Comme toute solution de séparation de l'identificateur et du localisateur, RANGI nécessitera le déploiement du nouveau système de correspondance entre identificateur et localisateur. Mais, contrairement à LISP, RANGI envisage de réutiliser le DNS.

À noter que le RRG insistait sur la déployabilité incrémentale des solution, nécessaire pour éviter d'arrêter tout l'Internet pendant les années de la transition. RANGI la permet mais les premiers adoptants ne gagnent rien à le faire, ce qui conduit à s'interroger sur la possibilité d'un déploiement effectif. Après tout, HIP, solution élégante et sûre, normalisée depuis longtemps, implémentée pour Linux et FreeBSD, n'a jamais décollé, en partie à cause de ce problème de la « récompense initiale ».

Proposition suivante, en section 4, IVIP ("*Internet Vastly Improved Plumbing*", dû à R. Whittle). Système de séparation cœur-bords, comme LISP, proche de LISP par beaucoup d'aspects, il s'en distingue notamment par le fait que les correspondances entre identificateurs et localisateurs sont poussées, plutôt que tirées, et que donc tous les routeurs d'entrée de tunnel ont un accès rapide à tout moment à une copie complète de la base. Cette mise à jour en temps réel élimine la nécessité pour les ITR (routeurs d'entrée des tunnels) de tester que l'ETR correspondant attend toujours les paquets à la sortie. En contrepartie, elle exige beaucoup de mémoire chez l'ITR, de l'ordre de dizaines de Go (cela semble énorme mais, d'ici à l'adoption massive, le moindre ordinateur de bureau aura autant de RAM que cela).

Cette base des correspondances, disponible en temps réel, malgré le taux de changement qu'imposent, par exemple, les machines et les réseaux mobiles, est évidemment la principale faiblesse d'IVIP. Certes, elle permet d'éliminer complètement plusieurs problèmes sérieux, concernant la mobilité ou la détection des pannes d'un ETR, mais est-elle réaliste ? Il n'y a aucun précédent développé et déployé avec succès... Le DNS ne fonctionne qu'en renonçant à la propagation instantanée des informations. DRTM ("*Distributed Real Time Mapping*", le système de correspondance d'IVIP) fera-t-il mieux ? En tout cas, il est intéressant de voir ce que permet cette approche.

Autre point à noter : les solutions de séparation du cœur et des bords, comme LISP et IVIP, sont parmi les seules à ne pas avoir besoin de noms uniques et donc à pouvoir fonctionner avec les petites adresses IPv4. La plupart des autres solutions dépendent d'IPv6.

Mais IVIP et LISP peuvent fonctionner avec IPv6. Au contraire, hIPv4 est spécifique à IPv4. Il a souvent été dit que la séparation entre identificateur et localisateur était déjà réalisée dans l'Internet, via le NAT, où l'adresse privée est un identificateur et l'adresse publique est un localisateur. C'est bien sûr largement une boutade (par exemple, l'adresse privée n'est pas unique, une machine qui se déplace peut changer d'adresse privée, etc), mais cela peut inspirer des idées qui ont des points communs avec le NAT comme hIPv4 (protocole créé par Patrick Frejborg). L'idée de base est que chaque machine a un identificateur sous forme d'une adresse IPv4 et un localisateur de même syntaxe. Les deux noms seront insérés dans chaque paquet (une petite couche entre la couche 3 et la couche 4), les routeurs de bord du domaine faisant l'échange des noms, pour que les autres routeurs n'aient pas à être modifiés. Du fait que la machine terminale doive insérer ces deux noms (l'identificateur et le localisateur), son système d'exploitation doit être modifié et hIPv4 est donc une des rares solutions qui nécessitent des changements à la fois dans les machines terminales et dans certains routeurs. Autre problème, l'épuisement des adresses IPv4 <https://www.bortzmeyer.org/epuisement-adresses-ipv4.html> fait qu'on peut se demander d'où viendront ces adresses... Ce problème, celui de la déployabilité, et la question (qu'on retrouve dans le NAT) des protocoles qui s'échangent directement des adresses IP (comme SIP) fait que hIPv4 fait partie des propositions qui sont explicitement rejetées.

Continuons, puisque l'imagination des participants au RRG est sans limites. La section 6 présente NOL ("*Name OverLay*"). Actuellement, dans l'Internet, les noms de domaine, gérés dans le DNS, sont complètement extérieurs à la machine elle-même. C'est au point qu'une machine n'a pas de moyen simple de connaître son propre nom de domaine (si elle en a un !) sans parler de l'idée de l'influencer ou de le changer. L'idée de base de NOL est d'ajouter une couche de gestion des noms de domaine à toutes les machines, et d'utiliser ces noms comme identificateurs. (Voir les "*Name-Based Sockets*", en section 15, une idée qui présente de nombreuses similarités.) Il est difficile de la décrire en détail car, apparemment, aucune spécification écrite n'a été produite. Il n'est pas nécessaire de modifier le système d'exploitation mais il faudra modifier les bibliothèques utilisées habituellement pour la résolution de noms (sur Unix, la libc). NOL n'implique pas de modifier l'application des deux côtés (contrairement à NBS, "*name-based sockets*", présenté plus loin) car des relais spéciaux traduisent les noms et les adresses IP (une sorte de routeur NAT étendu). NOL n'a pas bénéficié de critiques ou de jugements précis, peut-être en raison de son caractère trop exotique.

C'est seulement avec la section 12 qu'apparait ILNP ("*Identifier-Locator Network Protocol*", fait par R. Atkinson & S. Bhatti, site officiel en <http://ilnp.cs.st-andrews.ac.uk/>, bons transparents d'introduction en à NANOG <http://www.cs.st-andrews.ac.uk/~saleem/talks/2010/ilnp_nanog50/2010-10-03-ilnp_nanog50-v4_final.pdf>, RFC 6740 et suivants, l'architecture qui a finalement eu les faveurs des présidents du groupe. Il repose sur une stricte séparation de l'identificateur et du localisateur, avec le DNS pour passer de l'un à l'autre. Pour gérer les changements de connectivité, ILNP fait donc un usage intensif de DNSSEC et des mises à jour dynamiques du DNS (RFC 2136 et RFC 3007). L'auteur semble d'ailleurs prêter trop de pouvoirs à DNSSEC, par exemple en disant que les requêtes PTR peuvent également être authentifiées. C'est vrai qu'on peut signer un enregistrement PTR mais, lorsque le PTR de `2001:db8::dead:beef` pointe vers `www.amazon.com`, il n'y a aucun moyen de s'assurer qu'il en a le « droit », DNSSEC ou pas.

ILNP atteint plusieurs des objectifs du RFC 5887 mais pas tous. Ainsi, certains serveurs (ceux du DNS, par exemple) doivent toujours être décrits par un localisateur. Dans son état actuel, ILNP impose l'usage du DNS pour tout (plus de `ping 2001:db8:1:3::cafe`), ce qui ne convient pas forcément à toutes les applications. Il faudra donc sans doute étendre ILNP sur ce point.

Dernière architecture de routage envisagée dans notre RFC, IRON/RANGER (section 16) où IRON signifie "*Internet Routing Overlay Network*" (auteur : F. Templin, décrit dans le RFC 6179) et s'appuie sur l'architecture RANGER ("*Routing and Addressing in Networks with Global Enterprise Recursion*", RFC 5720). RANGER est dérivé de ISATAP (RFC 5214) et fournit un système de tunnels automatiques pour transporter des paquets identifiés par un... identificateur à travers un cœur qui n'utilise que des localisateurs. À son tour, RANGER s'appuie sur le mécanisme SEAL ("*Subnetwork Encapsulation and Adaptation Layer*", RFC 5320, qui gère notamment les questions de MTU, toujours cruciales <https://www.bortzmeyer.org/mtu-et-mss-sont-dans-un-reseau.html> avec les tunnels. La critique de ce mécanisme note que beaucoup de détails manquent, notamment la fonction de recherche de la correspondance entre un identificateur et un localisateur. Elle remarque aussi que SEAL empêcherait de déployer les jumbogrammes. Dans sa réponse à la critique, l'auteur note que IRON ne repose pas sur un système de "*mapping*" classique mais sur une combinaison du protocole de routage et d'un protocole spécifique de distribution des correspondances. Personnellement, cela me semble encore très flou.

Une autre architecture est présentée en section 14, Évolution. Conçue par Lixia Zhang, une vétéran de l'IETF, auteure de plus de trente RFC, Évolution propose une approche du passage à l'échelle du système de routage qui est assez originale. L'idée de base est de faire feu de tout bois, en agrégeant les routes partout où on peut, sans attendre le « Grand Soir » d'une nouvelle architecture. Évolution critique la tendance des autres propositions du RRG à considérer qu'on ne peut rien résoudre tant qu'on n'a pas changé tout le routage. Mais, comme le peu de déploiement de IPv6 l'a montré <https://www.bortzmeyer.org/ipv6-et-l-echec-du-marche.html>, les solutions qui aideront tout le monde, une fois que tout le monde les aura déployé, ont le plus grand mal à s'imposer sur l'Internet. Les solutions qui gagnent sont au contraire en général celles qui procurent un bénéfice immédiat et à court terme aux décideurs.

Évolution propose donc de commencer par le routeur seul : tout routeur pourrait déjà agréger massivement les routes qu'il connait, économisant ainsi sa mémoire. Cet effet est d'autant plus marqué que le routeur a peu de liens physiques possibles. Ensuite, au sein cette fois d'un système autonome, on pourrait désigner certains routeurs comme étant les seuls à avoir une connaissance complète des liens externes, la plupart des routeurs de l'AS pouvant alors se contenter de tunnels vers ces « super-routeurs », créant une sorte de schéma "*Map-and-Encap*" local. Une fois cette « agrégation virtuelle » réalisée à l'intérieur de plusieurs AS, ceux-ci pourraient annoncer ces super-routeurs via BGP, étendant leur bénéfice aux autres AS. Enfin, une séparation plus intense entre le contrôle des routes (fait par BGP, et qui limite l'agrégation puisqu'il faut pouvoir annoncer les routes aux autres) et la transmission effective des paquets (où le routeur peut se contenter d'une FIB bien plus petite que sa RIB) complèterait le dispositif. Le point important d'Évolution est que chaque étape apporte ses bénéfices, et qu'il n'est donc

pas indispensable de faire miroiter des bénéfices lointains aux décideurs pour les convaincre. Chaque étape a sa récompense propre.

Mais Évolution a ses propres problèmes. Certaines des techniques d'agrégation perturbent les systèmes existants (par exemple, l'agrégation de N préfixes en un super-préfixe, qui couvre un espace plus grand que ne le faisaient les N sous-préfixes, crée des adresses routables qui n'existaient pas avant, ce qui peut entraîner de nouvelles boucles de routage, et peut gêner les systèmes de sécurité comme RPF - RFC 3704). D'autre part, l'agrégation virtuelle allonge le chemin pris par les paquets, et peut aussi gêner RPF. À noter qu'Évolution ne fournit pas de solution pour la mobilité, concept à la mode, mais dont il n'est pas évident qu'il doive être géré par le système de routage (plutôt que, par exemple, une combinaison de DHCP et de meilleures applications).

Une autre critique faite à Évolution est plus philosophique : en fournissant des bénéfices à court terme, n'encourage-t-on pas les rustines sur un système déjà pas mal rapiécé, au risque de détourner les gens d'une transition, certes coûteuse mais nécessaire ?

On l'a vu, un problème commun à toutes les propositions de séparation du localisateur et de l'identificateur est le système de correspondance ("*mapping*") entre les deux. Cela a mené certains à concentrer leurs efforts sur ce problème et à travailler sur des systèmes de correspondance génériques, pouvant être utilisés pour plusieurs protocoles. C'est le cas de LMS ("*Layered Mapping System*"), en section 8. Comme le ALT de LISP, il repose sur une hiérarchie des noms (système de résolution bien plus éprouvé que les systèmes plats, qu'utilisent par exemple les DHT). Mais, contrairement à ALT, la correspondance entre noms est complètement déconnectée du routage : ce ne sont pas les FAI ou les opérateurs réseau qui font la résolution de noms.

Autre système de correspondance, le "*2-phased mapping*" de la section 9. Ce système vise à trouver l'ETR (le routeur de sortie du tunnel, celui qui est le plus « proche » du réseau visé) correspondant à un préfixe donné, le préfixe contenant l'identificateur de la machine visée. Tout serait simple si cette correspondance préfixe -¿ ETR était relativement statique. Mais, évidemment, ce n'est pas le cas et "*2-phased mapping*" considère qu'un système qui stockerait toutes les relations préfixe -¿ ETR de la planète ne tiendrait pas longtemps. Son idée centrale est d'intercaler un numéro de système autonome (AS) entre les deux, de façon à avoir une relation préfixe -¿ AS -¿ ETR. Les systèmes autonomes connaissent en effet certainement leurs clients et donc les ETR qui les desservent. Il ne reste donc qu'à créer un système pour que les AS puisse informer le système de résolution des préfixes qu'ils servent. On peut donc envisager de d'abord résoudre le préfixe en un numéro d'AS puis d'interroger ce dernier. Ce système en deux étapes (les résultats de chacune pouvant être mis en cache) passe certainement mieux à l'échelle. À noter que les deux étapes ne sont pas forcées d'utiliser les mêmes techniques (par exemple, la première pourrait être faite avec le DNS, ou bien avec un serveur centralisé, type whois, et la seconde avec une extension à BGP). L'idée n'a toutefois pas été assez développée pour être considérée sérieusement par le RRG.

On l'a vu, toutes les propositions soumises au RRG ne concernaient pas forcément une architecture de routage nouvelle, certaines présentaient une solution auxiliaire, qui allait aider les nouvelles architectures. C'est le cas des "*Name-based sockets*" de la section 15. L'idée de base est de ne manipuler, depuis les applications, que des noms `<https://www.bortzmeyer.org/programmation-orientee-nom.html>` et de laisser les couches basses gérer et stocker les adresses. Dans la proposition actuelle, ces "*name-based sockets*" permettraient aux applications d'initier et de recevoir des communications uniquement avec les noms. Il ne s'agit pas seulement de cacher les adresses IP dans une bibliothèque utilisée par l'application (comme le font, par exemple, les bibliothèques curl et neon pour les programmeurs C) mais de déplacer leur gestion bien plus bas, dans la pile IP elle-même, permettant ainsi des choses comme le changement d'adresse IP en cours de communication.

En quoi est-ce que cela aiderait le routage ? Eh bien, cela réduirait la nécessité de disposer d'adresses PI et transformerait l'adresse IP est un nom largement invisible, comme l'est l'adresse MAC aujourd'hui. La souplesse que cela apporterait permettrait, par exemple, d'agréger plus radicalement les préfixes. Les NBS ("*name-based sockets*") ont bien d'autres propriétés intéressantes comme de rendre le NAT largement inoffensif (si l'application ne voit jamais d'adresses IP, leur changement par le routeur NAT n'aura pas de conséquence).

Les NBS sont donc une solution situéee dans la machine terminale, sans changement des routeurs. Leur déploiement suppose donc leur adoption par les systèmes d'exploitaion (rappelez-vous qu'il ne s'agit pas d'une simple couche pour aider le programmeur, les NBS doivent interagir fortement avec IP et donc être dans le noyau). NBS est plus longuement documenté dans l'article de Vogt, « "*Simplifying Internet Applications Development With A Name-Based Sockets Interface*" `<http://christianvogt.mailup.net/pub/vogt-2009-name-based-sockets.pdf>` ».

Quelles sont les limites des NBS ? D'abord, certaines applications auront toujours besoin d'adresses IP, par exemple celles nécessaires au "*bootstrapping*" comme les serveurs DNS. Ensuite, un réseau ne pourra abandonner ses adresses PI que lorsque **toutes** ses machines auront migré vers NBS.

Où cela nous mène t-il ? Comme indiqué au début, aucune proposition n'a recueilli le consensus du groupe. La section 17 expose donc la seule recommandation des présidents du groupe. Ceux-ci exposent d'abord le contexte, celui d'un Internet faiblement coordonné, sans point de décision central, et où les changements ne peuvent être introduits que de manière incrémentale. Tout changement étant compliqué, il faut choisir avec soin les changements qu'on propose. L'Internet a tenu, et très bien, jusqu'à présent, grâce à un grand nombre de modifications partielles, faites au fur et à mesure que les problèmes se posaient. Mais l'accumumation de ces changements a fini par mener à un réseau excessivement complexe et difficiles à maintenir. Les présidents du RRG proposent donc à la fois des changements à court terme, pour traiter les cas les plus urgents et une « bonne » solution à long terme, dont ils reconnaissent qu'elle n'existe pas encore. La recommandation est donc de travailler sur trois techniques :

— À court terme, une agrégation plus vigoureuse des préfixes, comme proposé dans le plan Évolution (section 14 et "*Internet-Draft*" `draft-zhang-evolution`).
— À long terme, une nouvelle architecture fondée sur ILNP ("*Identifier-Locator Network Protocol*", section 12).
— À court et moyen terme, travailler à rendre la renumérotation des réseaux plus facile, pour diminuer la pression en faveur des adresses PI. Décrit dans le RFC 5887, cette question est désormais du ressort du futur groupe de travail Renum.

Pourquoi ces choix ? L'agrégation de Évolution parce qu'on a besoin de solutions à court terme. ILNP parce que c'est une solution architecturalement propre, qui sépare nettement identificateur et localisateur, et qui ne dépend pas de tunnels. Et la rénumérotation facilitée parce que, quelle que soit la solution finale choisie, on ne pourra pas éviter des rénumérotations de temps en temps.