RFC 7927 : Information-Centric Networking (ICN) Research Challenges

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 25 août 2016

Date de publication du RFC : Juillet 2016

https://www.bortzmeyer.org/7927.html

Quels sont les défis qui attendent l'ICN ("*Information-Centric Networking*"), ce paradigme où le composant de base du réseau n'est pas la machine ou l'interface réseau mais l'unité d'information ? Le concept est à la mode depuis pas mal d'années mais ne progresse guère sur plusieurs points importants. Ce RFC de synthèse fait le point sur ce qui ne marche pas ou pas parfaitement dans l'ICN, et les axes de recherche souhaitables.

L'idée de base de l'ICN (également nommé CCN pour **Content-Centric Networking**) est de considérer que le réseau sert surtout à accéder à du contenu (un point de vue Minitel 2.0 très contestable) et qu'il faut donc architecturer le réseau autour de cet accès au contenu. Une unité d'information a un **nom** et le réseau permet avant tout d'accéder à cette unité d'information, en échange de ce nom (`get (NOM)` ...) L'unité d'information est parfois nommée NDO pour "*Named Data Object*". On voit que la localisation de l'information (que ce soit dans l'espace physique, ou dans celui défini par la topologie du réseau Internet) n'est pas prise en compte. Cette approche permettrait, en théorie, une meilleure efficacité (la mise en cache par les composants du réseau améliorerait les performances), un meilleur passage à l'échelle (la réplication de données populaires serait simple), et une meilleure résilience (le contenu serait à plusieurs endroits). Cette idée est décrite plus longuement dans le RFC 7476 [1] (lecture recommandée, avant ce RFC 7927), et j'en ai déjà écrit une critique <https://www.bortzmeyer.org/van-jacobson-ccn.html>.

Pour la mettre en œuvre, on peut envisager une couche supplémentaire « accès au contenu » au-dessus de l'Internet actuel, ou bien une approche « table rase » où on remplace IP par un protocole orienté ICN. Notez qu'il existe déjà des mécanismes pour accéder à du contenu par son nom, en ne

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc7476.txt

se souciant pas de la localisation, mécanismes qui fournissent la rapidité, le passage à l'échelle et la robustesse dont on parlait plus haut. Le plus évident est bien sûr BitTorrent mais on notera que les promoteurs de l'ICN n'en parlent quasiment jamais...

D'abord, une synthèse des problèmes avec l'approche actuelle, vus du point de vue des partisans de l'ICN (section 2 du RFC). Cette approche, qualifiée de "*host-centric*", met au centre du réseau la machine. Identifiée par des noms de domaine et/ou des adresses IP, c'est l'unité de base du réseau. On peut bien sûr bâtir une couche ICN (ou para-ICN) au-dessus de cette approche (un CDN est un exemple partiel) mais, comme le réseau n'est pas au courant, cela entraine des limitations :

— La circulation du trafic n'est pas optimale, puisque l'infrastructure du réseau (le routage, par exemple) ne peut pas optimiser pour le trafic des unités d'information,
— Les capacités du réseau, par exemple en diffusion, ne sont pas ou peu utilisées,
— Il n'est pas évident pour le réseau de savoir ce qu'il peut garder dans un cache local (le Web doit recourir à des CDN explicitement configurés),
— La validation des données (la vérification de leur authenticité et de leur intégrité) n'est pas assurée par le réseau et doit utiliser d'autres mécanismes.

Pour ce dernier point, celui de la validation, je suis personnellement toujours stupéfait que le Web n'ait toujours pas un mécanisme standard pour authentifier une page qu'on a récupéré. À l'heure actuelle, il faut toujours se contenter de mécanismes ad hoc, comme une signature PGP détachée du fichier qu'on veut valider. (Non, HTTPS n'est pas une solution, d'abord parce qu'il ne fonctionne qu'en transfert, pas après, donc on ne peut pas valider quelque chose qui stationne dans un cache, et ensuite parce qu'il est incompatible avec des miroirs, sauf à distribuer une clé privée à plein d'autres acteurs.)

Avant d'aller plus loin, la section 3 du RFC nous apporte une petite révision des termes et concepts essentiels en ICN. Pour les termes, il faut se souvenir de :

— NDO ("*Named Data Object*") : une unité de données ayant un nom,
— Demandeur ("*requestor*") : l'entité qui demande un NDO, en indiquant son nom,
— Éditeur ("*publisher*") : l'éditeur qui publie un NDO, le met en ligne. L'éditeur n'est pas forcément le créateur du NDO.

Une fois armé de cette terminologie, on peut expliquer le concept central de l'ICN : un réseau dont l'objet de base est le NDO (pas le paquet, pas la machine), et dont le service de base est d'envoyer des NDO aux demandeurs.

Ceux qui ont assisté à mon exposé à Pas Sage En Seine 2016 <https://www.bortzmeyer.org/pas-sage-en-seine-ethereum.html> ont pu voir un exemple d'ICN mis en œuvre sur la chaîne de blocs (exemple `FindByHash`).

Sur le papier, c'est très joli, mais voyons maintenant les défis que cela pose (section 4 du RFC, le gros morceau de ce texte). D'abord, le nommage et l'authentification (les deux sont en général liés dans l'ICN). Nommer un NDO est aussi crucial dans l'ICN que de donner une adresse IP à une machine dans l'Internet classique. Et comme l'objet nommé peut être récupéré à partir de divers endroits dans le réseau ICN, vérifier l'authenticité par le biais de l'origine marche encore moins que dans le Web actuel. Il y a des tas de solutions possibles, résumées dans des études comme « "*Naming in Content-Oriented Architectures*" <https://people.eecs.berkeley.edu/~alig/papers/content-oriented-naming.pdf> » de Ghodsi, A., Koponen, T., Rajahalme, J., Sarolahti, P., et S. Shenker ou bien « "*A Survey of Information-Centric Networking*" <http://drops.dagstuhl.de/opus/volltexte/2011/2941/pdf/10492.KutscherDirk.Paper.2941.pdf> » de Ahlgren, B., Dannewitz, C., Imbrenda, C., Kutscher, D., et B. Ohlman.

Il y a deux grandes façons d'organiser le nommage dans l'ICN : espace de nommage hiérarchique, ou bien espace de nommage plat. Par exemple, un nommage hiérarchique pourrait produire des noms

comme `pays/éditeur/date/NDO`. Cette façon de nommer simplifie nettement le routage et le passage à l'échelle. La structure des noms permet de faire une structure identique pour la PKI qui servira à authentifier le contenu (ainsi, une AC pour un éditeur donné ne pourra pas signer des certificats d'un autre éditeur, contrairement à ce qu'on voit sur le Web aujourd'hui, où la structure arborescente des noms de domaine n'est pas utilisée pour limiter les dégâts que peut faire une AC.) Cette structure hiérarchique peut aussi permettre de faire des noms lisibles par des humains `<https://www.bortzmeyer.org/beaux-urls.html>`. (Par contre, le RFC oublie de noter que de tels noms suscitent d'innombrables disputes de gouvernance : tout le monde veut être à la racine, veut tel nom, etc.)

Autre solution, l'espace de nommage plat. Typiquement, le NDO est désigné par un condensat cryptographique de son contenu. Cela permet l'authentification de manière évidente : on récupère le contenu, on calcule le condensat, et on doit retrouver le nom. Et un tel système peut être complètement réparti, sans autorité jouant un rôle particulier. Par contre, l'absence de structure dans le nom nécessitera un système de résolution adapté (comme une DHT). Une telle solution est même déjà décrite dans un RFC, le RFC 6920.

Aucune des deux façons de gérer l'espace de nommage n'est parfaite (cf. le triangle de Zooko). Par exemple, les condensats cryptographique ne sont pas maniables par un humain, mais les noms hiérarchiques ne peuvent pas être alloués de manière complètement pair-à-pair.

Les sujets de recherche existants sur le nommage sont :

— Le nommage d'objets qui n'existent pas encore (un flux vidéo filmé en temps réel, par exemple). On ne peut donc pas utiliser de condensat cryptographique. Mais peut-être un mécanisme à deux étapes, un structure de données non-cryptographique décrivant le contenu, qui serait elle-même condensée ?
— La protection de la vie privée. Avec de l'ICN ordinaire, tout le monde voit à quel contenu vous accédez. Cela concerne même les équipements intermédiaires, puisque le but est justement que le réseau voit tout.
— Nommer les nouvelles versions d'un objet. Un principe de base de l'ICN est que le nom désigne le contenu. On change une virgule à un texte, il faut un nouveau nom (c'est particulièrement évident pour les noms fondés sur un condensat cryptographique). Comment gérer la notion de « nouvelle version, légèrement modifiée » ? Une possibilité serait d'avoir des noms incluant un numéro de version, ce qui nécessiterait un mécanisme de résolution adapté (pour des opérations comme « je veux la dernière version »).
— Restreindre l'accès dans certains cas. Le paradigme de l'ICN est que tout le monde a accès, ce qui permet aux éléments du réseau eux-même d'avoir accès à tout et, par exemple, de le mettre en cache. Si on veut des NDO à accès restreint, comment faire ? Les solutions de sécurité du monde réel, comme le chiffrement de bout en bout sont incompatibles avec le paradigme de l'ICN.

En parlant de restriction d'accès, la sécurité, dans son ensemble est un difficile problème pour l'ICN (section 4.2). C'est le cas de la plupart des solutions miracles qui sont proposées pour l'Internet : une idée simple qui semble bien marcher, puis, dès qu'on creuse les détails pratiques, les ennuis surgissent puis, dès qu'on arrive à la sécurité, l'idée simple s'effondre. Dans le cas de l'ICN, le RFC suggère plusieurs pistes de travail. D'abord, l'authentification des données, sans doute la partie la plus facile à réaliser (et celles, on l'a vu, où le Web actuel est défaillant). Comme on peut récupérer un NDO depuis n'importe où, pas forcément depuis le serveur d'origine, il est crucial de l'authentifier. Évidemment, si le nom de l'objet est un condensat de son contenu, comme c'est le cas avec le RFC 6920, la vérification de l'intégrité est facile : on récupère l'objet, on condense et on regarde si on tombe bien sur la même valeur. Pour des objets générés dynamiquement, cette solution ne fonctionne pas et il faut donc signer l'objet avec de la cryptographie asymétrique. D'ailleurs, cette cryptographie est nécessaire dans tous les cas : vérifier le nom-qui-est-un-condensat prouve l'intégrité de l'objet, pas que l'objet vient bien de l'auteur attendu. Ici, une signature cryptographique est donc également nécessaire. Comme toujours avec la cryptographie à clés publiques, cela laisse ouvert l'énorme problème de la distribution des clés... (cf. RFC 5280 pour une approche possible).

Pour l'utilisateur normal, vérifier que l'objet `148aec5042e7c05df755d9ce8adec80a1e9c10b1557194fd94e` a bien un contenu qui correspond à ce condensat n'a qu'un intérêt limité. M. Michu voudrait une authentification plus forte, vérifier qu'il s'agit bien du guide de sécurité Java de l'ANSSI <`https://www.ssi.gouv.fr/uploads/2013/04/np_securiser_jre_notetech.pdf`> ou que c'est bien la série télé qu'il cherchait <`http://www.ina.fr/PackVOD/PACK883659600`>. Et, là, le problème est difficile. Les défenseurs du pair-à-pair prétendent par exemple souvent que BitTorrent est pair-à-pair, qu'il fonctionne sans centre mais, pour la très grande majorité des usages, c'est faux. M. Michu utilise un moteur de recherche centralisé, comme The Pirate Bay ou comme isoHunt. Il récupère alors un "*magnet*" (qui contient un condensat du fichier convoité), et, ensuite, on est vraiment en pair-à-pair : on télécharge, on vérifie le condensat et on s'installe dans son fauteuil pour regarder. Mais l'étape du moteur de recherche, avec ses pubs mensongères, était tout, sauf pair-à-pair.

En ICN, comment lier un NDO à une identité du monde extérieur, par exemple un acteur connu et de confiance ? C'est un problème essentiel mais non encore résolu.

Le contrôle d'accès, on l'a vu, est également un problème difficile. Comment faire si je veux vendre du contenu sans permettre aux destinataires de le redistribuer ? Il n'y a pas de solution évidente. Les approches intégrées sont celles où l'entité qui publie utilise des mesures pour limiter la redistribution (typiquement des menottes numériques), les approches séparées comptent sur une tierce-partie qui gère les accès (je vous laisse imaginer les conséquences en terme de liberté de choix de son logiciel par M. Michu...) On peut bien sûr chiffrer le contenu qu'on va distribuer (ah, Canal+...) mais il reste à distribuer les clés ensuite.

La cryptographie n'est pas éternelle, les algorithmes de cryptographie finissant par être affaiblis, puis cassés. Ce n'est pas un problème si on publie du contenu à courte durée de vie mais, si le contenu est encore utile et payant dans dix ans, l'efficacité de la cryptographie devient incertaine : comment garantir que l'algorithme utilisé sera toujours incassé ? L'idéal serait un mécanisme de re-signature et de re-chiffrement mais cela pose encore le problème de la gestion des clés privées.

Autre conséquence de la mise en cache et de la distribution à partir de plusieurs endroits, l'auteur d'un contenu n'aura plus de statistiques d'accès fiables. De la même façon, retirer un contenu publié sera encore plus difficile que sur le Web actuel (un contenu populaire est davantage répliqué et donc plus dur à supprimer).

Autre problème de sécurité avec l'ICN, le risque d'attaque par déni de service via le cache. Un des buts de l'ICN est de permettre, et même d'encourager, la mise en cache automatique du contenu par les éléments du réseau, comme les routeurs. Cela ouvre une possibilité d'attaque où le méchant téléchargerait plein de choses sans intérêt juste pour remplir les caches. Pire : comme l'ICN renonce explicitement au principe de bout en bout, ces éléments intermédiaires mantiennent un état et peuvent également être attaqués par ce biais. Par exemple, en annonçant plein de contenus divers, on pourrait remplir la table de routage.

Après la sécurité, le routage dans l'ICN pose également des défis intéressants. Le terme n'a pas tout à fait le même sens dans l'ICN que dans l'Internet actuel. L'ICN permet d'obtenir un contenu (le NDO, "*Named Data Object*") en échange d'un nom. Cela nécessite trois étapes : la première est de résoudre le nom en un localisateur, plus concret. La deuxième est d'envoyer la demande de contenu à cet endroit. La troisième est d'obtenir la réponse (le NDO). Ces trois étapes peuvent être faites de différentes manières, catégorisées comme **routage par nom** ("*route-by-name*"), **recherche par nom** ("*lookup-by-name*") et **hybride**.

Le routage par nom consiste à ne pas avoir la première étape, celle de résolution, ce qui simplifie le modèle : le réseau comprend directement les noms. Il faut donc une « table de routage » dont la taille

permette de stocker tous les NDO. Comme on envisage entre 10{}15 et 10{}22 NDO, le défi technique est colossal. Sans compter qu'il faut aussi identifier la source de la requête, puisqu'il faudra bien lui envoyer le contenu demandé (c'est ce que fait l'algorithme des miettes de pain, cf. Rosensweig, E. et J. Kurose, « "*Breadcrumbs : Efficient, Best-Effort Content Location in Cache Networks*" <http://ieeexplore.ieee.org/document/5062201/> »). Les sujets de recherche sur le routage par nom comportent donc, entre autres :

— Les mécanismes d'agrégation permettant de réduire le nombre d'entrée dans la table de routage,
— Si on utilise des noms hiérarchiques (par exemple `/IETF/IRTF/ICN/Research challenges` pour ce RFC) afin de permettre l'agrégation, comment permettre aux utilisateurs d'utiliser des noms plus simples ? (Se servir d'un moteur de recherche serait de la triche, puisqu'on voulait se dispenser de l'étape de résolution.)

La catégorie « recherche par nom » regroupe les mécanismes de routage qui, eux, ont une étape de résolution explicite. On résout le nom (qui est pratique pour un humain, et stable <https://www.bortzmeyer.org/pourquoi-le-dns.html>) en un localisateur (qui est pratique pour le réseau et qui pourrait être une simple adresse IP). C'est par exemple ainsi que fonctionne le système « "*MDHT : A hierarchical name resolution service for information-centric networks*" <http://conferences.sigcomm.org/sigcomm/2011/papers/icn/p7.pdf> ». Les défis techniques à relever sont l'accès rapide au localisateur (en se rappelant que le contenu peut être à plusieurs endroits) et la mise à jour efficace (car le contenu peut changer de localisation, c'est même un des buts de l'ICN). Notez que les gens de l'ICN aiment bien réinventer la roue et que le DNS n'est même pas mentionné.

Enfin, il y a les solutions de la catégorie hybride. Par exemple, on fait du routage par le nom en local mais on résout le nom en un localisateur dès qu'on sort de ce domaine local.

Vous aimez les défis techniques et la liste n'est pas encore assez longue pour vous ? Il reste plusieurs problèmes à affronter. Par exemple, le contrôle de congestion. Il n'a pas à être de bout en bout puisque, dans l'ICN, la communication peut se faire via des éléments intermédiaires (voir par exemple « "*ConTug : A Receiver-Driven Transport Protocol for Content-Centric Networks*" <http://users.piuha.net/blackhawk/contug/contug.pdf> »).

On a vu qu'un des intérêts principaux de l'ICN était de permettre la mise en cache automatique par des éléments du réseau, et l'accès au contenu à partir de ces caches. La bonne utilisation de ceux-ci soulève plusieurs questions :

— Placer ces caches sur le chemin des données ("*on-path caching*"), ou bien à côté, avec une redirection, comme avec les CDN actuels ("*off-path caching*") ? Le cache sur le chemin est plus conforme aux principes d'architecture de l'ICN, mais il suppose des équipements capables de suivre tout le trafic, donc très rapides. Et ces équipements seront-ils rentabilisés ? Bien des contenus sont peu accédés et le cache ne servira donc pas autant qu'on ne pouvait l'espérer.
— Combien d'équipements capable de gérer un cache faut-il placer ?
— Comment gérer l'obsolescence ? Un contenu populaire va se trouver stocké dans de nombreux caches distribués un peu partout. Lorsque le contenu est modifié, ces caches auront une information obsolète. Il y a plusieurs approches pour ce problème. On les classe en général en approches directes, où l'information est stockée dans le NDO lui-même (par exemple une date d'expiration, après laquelle il faudra virer le contenu du cache), et approches indirectes, où on re-valide la fraicheur du contenu auprès du serveur original (ce qui suppose que son identité soit marquée dans le NDO). Les approches indirectes permettent de gérer le cas où la date d'expiration n'est pas connue à l'avance. Notez qu'un système où le nom change avec le contenu (comme les `ni:` du RFC 6920) ne résout pas le problème : il faut maintenant un mécanisme pour informer les clients du nouveau nom.

Les administrateurs réseau sont habitués à des outils comme ping et traceroute pour tester et déboguer leur réseau. Comment feront-ils avec l'ICN ? C'est un autre défi. Étant donné que ping et traceroute ne sont apparus que de nombreuses années après IP, on peut prévoir que, pendant un certain temps, les administrateurs des réseaux ICN souffriront du manque d'outils adaptés.

Quelles applications utiliseront l'ICN, si tous ces défis sont relevés, et avec succès ? Le RFC se termine par une liste d'applications potentielles (cf. RFC 7476) :

— Le Web semble un candidat évident, le paradigme « tout n'est qu'accès à du contenu » collant à peu près à une partie des usages du Web (en gros, les utilisations de la méthode HTTP `GET`). Les noms des NDO remmplaceront-ils les URI ? Il n'est pas évident que tous les usages du Web (`POST`, `PUT`, et autres méthodes utilisées par les applications REST) s'adaptent aussi bien à l'ICN.
— La vidéo est typiquement un domaine où la pure consommation (on se contente d'accéder à du contenu) est très répandue. Elle semble donc un autre bon candidat pour l'ICN. Un autre document du groupe de recherche ICN de l'IRTF, le RFC 7933, étudie plus en détail cette application.
— Et l'Internet des Objets ? L'article de Baccelli, E., Mehlis, C., Hahm, O., Schmidt, T., et M. Waehlisch, « "*Information Centric Networking in the IoT : Experiments with NDN in the Wild*" `<http://conferences.sigcomm.org/acm-icn/2014/papers/p77.pdf>` » étudie ses liens avec l'ICN. Par exemple, bien des applications IoT reposent sur un modèle "*PUSH*", où l'objet envoie des données à son rythme, pas en réponse à une requête. Un tel modèle ne s'adapte pas bien à l'ICN.

Une conclusion personnelle ? Il y a plein d'idées intéressantes dans l'ICN, et c'est donc un sujet de recherche utile, malgré la prémisse de départ (« tout est accès au contenu ») qui est fausse.