Nommage : avoir le beurre, l'argent du beurre, et le sourire de la crémière

Stéphane Bortzmeyer
AFNIC
bortzmeyer@nic.fr

Alcatel-Lucent, 13 septembre 2011

Résumé

Explorer les systèmes de nommage utilisables sur l'Internet

En mettant l'accent sur leurs propriétés, et la difficulté à avoir toutes les propriétés favorables en même temps

Noms et adresses, définition académique

1. Un nom identifie un objet, et ne change pas, notamment en cas de déplacement
2. Une adresse identifie une « position » (dans quel espace ?) de l'objet et peut donc changer

A name indicates what we seek. An address indicates where it is.

(Jon Postel, RFC 791.) Mais voir le RFC 6115, section 1.2, pour une discussion plus moderne et plus réaliste. Ces deux définitions ne correspondent pas du tout à la réalité aujourd'hui.

Noms et adresses dans l'Internet aujourd'hui

La réalité, aujourd'hui, est :

1. Un nom est composé d'une série de caractères alphabétiques choisis par un utilisateur
2. Une adresse est composée d'une série de chiffres, choisis par une autorité distincte de l'utilisateur. Elle est en général utilisée pour indiquer une position (dans quel espace ?) aussi bien que comme identificateur.

Noms et adresses dans cet exposé

1. Pour désigner un objet de manière non-transitoire, j'utiliserai plutôt le terme « identificateur », pour éviter le problème.
2. Avec « nom » comme synonyme (mais ambigu).
3. Et « adresse » pour les identificateurs de bas niveau, obtenus après résolution d'un nom (attention, la résolution peut être récursive)

Dans ce sens, une adresse IP *Provider-Independent* est un nom.
Idem pour un ISBN pour un livre.

Propriétés souhaitées

On voudrait que les noms soient :

1. Uniques (imaginez le Web s'il y avait deux `www.facebook.com`...)

Propriétés souhaitées

On voudrait que les noms soient :

1. Uniques
2. Stables (pensez à un URL dans un article...)

Propriétés souhaitées

On voudrait que les noms soient :

1. Uniques
2. Stables
3. Parlants (`www.afnic.fr` plutôt que `2001:660:3003:2::4:20`)

Propriétés souhaitées

On voudrait que les noms soient :

1. Uniques
2. Stables
3. Parlants
4. Résolvables (le nom ne sert pas qu'à identifier, on veut trouver une adresse)

Propriétés souhaitées

On voudrait que les noms soient :

1. Uniques
2. Stables
3. Parlants
4. Résolvables
5. Obtenables (facilement, gratuitement, sans payer des sommes démesurées)

Propriétés souhaitées

On voudrait que les noms soient :

1. Uniques
2. Stables
3. Parlants
4. Résolvables
5. Obtenables
6. Sûrs (un méchant ne doit pas être facilement capable de détourner un nom à moi)

Réalisme

Peut-on avoir tout cela à la fois ?

Probablement pas (le triangle de Zooko). Il va donc falloir faire des compromis.

Pour appuyer l'assertion « on ne peut pas tout avoir à la fois », prenons quelques exemples.

Noms de domaines

`www.bortzmeyer.org`

1. Uniques grâce à l'allocation décentralisée
2. Assez stables (mais risques juridico-financiers)
3. Très parlants (c'est bien pour cela qu'ils suscitent des convoitises)
4. Très facilement résolvables, technologie fiable et éprouvée (DNS)
5. Payants, nécessitent de passer par des organismes qu'on n'aime pas forcément (cf. Laurent Chemla, « Confessions d'un voleur »)
6. Pas très sûrs mais on y travaille (sécurité de l'enregistrement, DNSSEC)

Host Identifiers

1. Créés par le protocole HIP (*Host Identity Protocol*, RFC 5201). Le HI (*Host Identifier*) est une clé cryptographique publique, le HIT (*Host Identity Tag*) son condensat.
2. Quasi-uniques, comme souvent en cryptographie (clés SSH ou PGP),
3. Très stables (tant qu'on ne perd pas la clé privée)
4. Pas du tout parlants (binaire pur). Le HIT est un peu plus gérable que le HI.
5. Pas de mécanisme de résolution universel (mais des essais)
6. Gratuits et générés entièrement localement, pas de registre à payer
7. Très sûrs (la machine signe ses paquets avec la clé privée de son nom...)

ISBN

1. Uniques grâce à l'allocation décentralisée
2. Très stables
3. Pas du tout parlants
4. Pas du tout résolvables (aucun service standard aujourd'hui, c'est encore Google qui est le plus efficace)
5. Pas de sécurité particulière (pas forcément utile)

ARK

1. Uniques grâce à l'allocation décentralisée
2. Identifiant stable choisi par la BNF
3. Pas parlant, exprès (un identificateur parlant empêche de changer d'avis, exemple `http://blog.example.org/machin-est-un-con.html` ne peut plus être modifié sans faire des 404)
4. Résolvable par une astuce : syntaxe standard pour faire un URL à partir d'un ARK. L'ARK `12148/bpt6k101412s` devient l'URL http ://catalogue.bnf.fr/ark :/12148/bpt6k101412s. Si `catalogue.bnf.fr` disparait, on peut remplacer par un autre.

[Troll] Mots-clés Google

On peut constater tous les jours que beaucoup de gens utilisent les mots-clés dans une recherche Google comme identificateur. « Pour voir le site d'Alcatel, tapez "Alcatel" dans Google »

1. Pas uniques, loin de là.
2. Aucune stabilité, change tous les jours (changements des « concurrents », changement de l'algorithme)
3. Très parlant
4. Résolvable très rapidement aujourd'hui, grâce à l'excellente infrastructure de Google

Exemples de conflits

1. L'unicité implique un système centralisé ou arborescent. Elle s'oppose donc au désir d'avoir des noms obtenables localement.
2. Le caractère parlant d'un nom s'oppose à sa stabilité (conflit de possession sur les noms de domaine : tout le monde veut `sex.com` alors que personne ne me réclame `2001:660:3003:2::4:20`)

Comprendre le conflit 1/2

Ne pas mélanger enregistrement et résolution

Ceux qui réclament « un DNS pair-à-pair » confondent souvent ces deux opérations. Il existe déjà des techniques pour la résolution en P2P, comme CoDoNS. L'enregistrement est une autre histoire.

Enregistrement en pair-à-pair

Préserver l'unicité en pair-à-pair est un défi. Exemple d'algorithme (dû à Emin Gün Sirer) :

1. Il y a N organisations d'enregistrements. dans `.example`
2. Pour tout nouveau nom, l'organisation qui l'enregistre signe un condensat du nom et passe à l'organisation suivante.
3. Chacune des N signe la signature précédente.

Comme la signature portait sur le condensat, personne ne peut tricher et voir quel nom était enregistré. Une fois que tout le monde a signé, l'enregistreur peut prouver qu'il était le premier. Pas de registre unique et on a quand même des noms uniques.

Exercice : trouver les défauts de ce système.

Comprendre le conflit 2/2

1. Plus haut, j'avais cité certains identificateurs comme « non résolvables ».
2. Il faut en fait nuancer. On peut résoudre n'importe quel identificateur, même plat (sans structure), avec :
   - 2.1 Un serveur centralisé
   - 2.2 Une DHT

Une solution parfaite ?

1. Namecoin est prometteur. Le principe est le même que Bitcoin : une chaîne de signatures (avec preuve de travail) et la chaîne la plus longue gagne.
2. Peu de détails publiés
3. Bien trop récent pour être sérieusement évalué

Un peu de théorie

L'impossibilité d'avoir toutes les propriétés à la fois est-elle prouvée ?

Pas au sens mathématique. Disons que c'est une conjecture.

Conclusion

1. Aujourd'hui, on ne peut **pas** avoir toutes les propriétés à la fois.
2. Demain, avec les progrès de la recherche fondamentale, ce sera peut-être possible **mais** c'est un problème **dur**. Chercheurs, vous êtes prévenus !