Séparation de l'identificateur et du localisateur

Stéphane Bortzmeyer
AFNIC
bortzmeyer@nic.fr

3 juillet 2007

Les rôles d'un identifiant sur le réseau

Du temps de IEN19, tout était simple.

Un identifiant peut servir à :

- Identifier une entité (*name*)
- Identifier un endroit où elle se trouve (*address*)
- Identifier un moyen d'y aller (*route*)

Mais l'Internet ne fonctionne pas comme cela !

TCP stocke les adresses dans sa TCB, pas les noms. Les adresses PI n'indiquent pas où on se trouve.

Et même IEN19 pointait déjà les limites du modèle.

Les qualités d'un identificateur

- mémorisabilité (un URI SIP est plus mémorisable qu'un numéro de téléphone et c'est pour cela qu'ENUM n'a pas d'avenir),
- permanence (au moins relative),
- efficacité pour le routage, donc taille fixe, résolution rapide.

Ces demandes sont souvent contradictoires

Par exemple, l'efficacité pour le routage demande une allocation hiérarchique d'adresses PA. La permanence demande une allocation d'adresses PI, d'où lutte des classes entre opérateurs et utilisateurs, avec les RIR qui hésitent.
Addresses can follow topology, or topology can follow addresses. Pick one.

Le protocole TTAP

Un exemple d'architecture à un seul identifiant : le protocole TTAP (*Twenty-Two Asnières Protocol*), alias le vieux téléphone d'autrefois.

1. Un seul identifiant, pour le réseau et pour les utilisateurs,
2. Très lié à la géographie (MONtigny 83 40),
3. Pas de portabilité, pas de permanence, la topologie se reflétant dans l'identifiant.

Le téléphone ne fonctionne plus ainsi : avec SS7 et la portabilité, le numéro de téléphone est d'avantage un nom qu'une adresse.

Architecture de l'Internet aujourd'hui

Deux identifiants :

- noms de domaines
- adresses IP

Ou plus si on regarde de plus près (labels MPLS, adresses MAC).

Dès qu'il y a plus d'un identifiant, il faut une correspondance (*mapping*) entre les deux.

C'est le rôle du DNS. BGP, ND ou LDP font aussi du *mapping*.

Les limites

- mobilité
- *multihoming*

La mobilité se gère en général avec DHCP. Problème : les sessions TCP n'y survivent pas et on dévoile à son correspondant sa localisation.

Le *multihoming* se gère en général avec BGP. Problème : on injecte une route de plus dans une table déjà bien pleine. Les ASIC des routeurs de la DFZ ont du mal à suivre.

Alignement des coûts et des bénéfices

Dois-je vraiment acheter de la RAM pour mon Juniper parce qu'une université thaïlandaise veut être multihomée ?

Toutes les disputes autour des adresses PI doivent s'interpréter dans ce cadre.

Une solution radicale

Utiliser le nom de domaine dans les sessions TCP.

Ainsi, l'adresse IP n'apparaitra que sur le câble.

La mobilité serait gérée en mettant à jour le DNS.

Même chose pour le *multihoming*.

Questions : est-ce réaliste de généraliser la mise à jour dynamique du DNS ? Faut-il acheter des actions Nominum ou Infoblox ?

Cela nécessite de modifier TCP partout... Et `netstat` ne monterait plus que des noms de domaine, jamais d'adresses IP.

À noter que SCTP permet partiellement cela.

Séparation

All problems in computer science can be solved by another level of indirection.

Idée dominante à l'IETF en ce moment : séparer l'adresse IP en un **identificateur** (qui garderait la stabilité) et un **localisateur** (qui garderait l'efficacité pour le routage).

1. L'un des gros enjeux est le *mapping* entre les deux.
2. Qui doit faire le nouveau travail, TCP, IP ou un *shim* entre les deux ?
3. L'importance de la sécurité.
4. La question des API.

Le protocole SSSP

Un exemple de séparation id / loc : le protocole SSSP (*Simple Stupid Separation Protocol*).

Présentation d'un protocole imaginaire mais qui permet de bien expliquer les enjeux et les choix.

SSSP en détail

1. SSSP est mis en œuvre dans TCP, il faut donc modifier les OS. Une autre solution aurait été d'intercaler un *shim* entre IP et le protocole de couche 4.
2. SSSP a des identificateurs plats, sans hiérarchie. Leur forme physique est celle d'une adresse IPv6, pour ne pas changer les API. On peut les générer avec ULA ou ORCHID.
3. SSSP garde les adresses IP (v4 ou v6) et les routeurs comme aujourd'hui. On peut déployer petit à petit. Les adresses redeviennent de purs localisateurs, elles sont toutes PA et changent donc souvent. Elles sont presque invisibles aux utilisateurs, un peu comme les adresses MAC aujourd'hui.
4. les identificateurs sont trouvés dans le DNS, avec le type AAAA,
5. les localisateurs sont trouvé dans une DHT (*Distributed Hash Table*).

Les forces de SSSP

SSSP gère bien la mobilité (il suffit de changer la correspondance entre identificateur et localisateur dans la DHT).

SSSP gère bien le *multihoming* (il suffit d'avoir plusieurs localisateurs par identificateur, ce que permettent les DHT typiques).

DHT en cinq minutes

Une DHT a deux opérations, get (qui prend une clé) et put (qui prend une clé et une valeur).

Les clés sont les identificateurs et les valeurs des localisateurs.

Je cherche à écrire à 4123:DEAD:BEEF::1, un get de cet identificateur me renvoie le localisateur 192.0.2.127.

Je suis 4123:DEAD:BEEF::1, je change de réseau, je fais un put du nouveau localisateur pour mon identificateur.

Avec OpenDHT :

```
% ./get.py 4123:DEAD:BEEF::1
192.0.2.127

% ./put.py 4123:DEAD:BEEF::1 10.42.1.3
Success
```

Les limites de SSSP

La principale est la sécurité. Comment authentifier les accès à la DHT ?

Pour un accès ultérieur, il est facile de fournir un secret (OpenDHT le permet).

Mais pour le premier accès ? La platitude de l'espace des identificateurs interdit les solutions hiérarchiques comme DNSSEC.

Les propositions sérieuses

SSSP ne peut donc pas être pris au sérieux.

Les solutions réalistes sont toutes bien plus complexes.

HIP

Host Identity Protocol, RFC 4423

Avec HIP, comme avec SSSP, l'adresse IP ne serait plus qu'un identifiant « technique », ne servant qu'à localiser la machine,

Le nouvel identificateur, le HI (*Host Identifier*) sera une clé publique cryptographique, qui sera allouée hiérarchiquement par PKI ou de manière distribuée par tirage au sort (comme le sont les clés SSH aujourd'hui).

La correspondance entre un HI et le localisateur (l'adresse IP) est apprise en lisant le premier paquet (et peut être vérifiée car elle est signée).

Pour l'initiateur de la connexion, il faut utiliser une méthode de « rendez-vous » pas encore spécifiée.

Bien que l'un des plus avancés (et disposant d'une implémentation), HIP n'a guère été testé au feu.

`http://hip.piuha.net/`

LISP

Locator/ID Separation Protocol

Actuellement à l'état de projet dans draft-farinacci-lisp.

Reporte cette séparation plus loin dans le réseau et ne nécesste donc aucune modification des *hosts*.

Les paquets IP utilisant les identificateurs sont encapsulés dans des tunnels utilisant les localisateurs.

SHIM 6

Pas encore de RFC. Voir
`http://www.ietf.org/html.charters/shim6-charter.html`

Utilise **une** des adresses IP de la machine comme identificateur (ULID).

Un *shim* entre IP et le protocole de transport note les adresses IP de la machine (toutes sont PA) et on peut en changer.

Ne gère que le *multihoming*, pas la mobilité (toutes les adresses possibles doivent être connues au début de la session).

Implémentations toujours en alpha.

SCTP

Stream Control Transmission Protocol, RFC 3286

Un concurrent de TCP qui, entre autres, fournit la possibilité de travailler avec plusieurs adresses IP.

Contrairement à shim6, c'est la couche 4 qui fait le travail.

Largement implémenté (Linux, Solaris et FreeBSD en production). Par exemple, Wireshark décode SCTP.

Références

- IEN19 *A note on inter-network naming, addressing and routing*. John Schoch. Le plus ancien texte conceptuel sur la question, janvier 1978.
- `draft-iab-raws-report`, compte-rendu du séminaire IAB sur le routage et l'adressage (*Routing and Addressing Workshop*) d'octobre 2006. Ce séminaire a notamment planché sur les limites des routeurs, les situant à environ cinq ans.
- `draft-carpenter-idloc-map-cons`, excellent texte de synthèse sur la séparation ID/Loc et notamment sur le problème du *mapping* entre identificateur et localisateur.