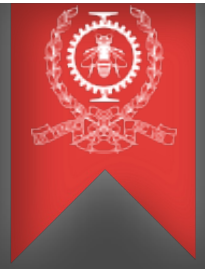


Service de répertoire de noms

Module 7

INF8480 Systèmes répartis et infonuagique

Michel Dagenais

École Polytechnique de Montréal
Département de génie informatique et génie logiciel

Problématique

- Nom d'ordinateur, d'utilisateur, de service sous forme textuelle à traduire en identificateur binaire et éventuellement en identificateur de bas niveau (nom/adresse IP/adresse Ethernet, nom de fichier/capacité/serveur-fichier).
- Base de donnée de paires nom-attributs.
- Service de consultation, recherche, découverte, modification, effacement, enregistrement.
- Serveurs hiérarchiques, répliqués.
- Doit fonctionner à l'échelle planétaire (nom des ordinateurs, utilisateurs...).
- Exemples: DNS, X500, CORBA Naming Service, Portmap, LDAP.

Organisation hiérarchique de noms

- Domaine de nom: niveau dans la hiérarchie pour lequel une autorité gère les noms mais peut déléguer un sous-domaine.
- Pour chaque nom une série de paires (type, valeur) est stockée: (usager, nom/téléphone/adresse...), (répertoire, liste de noms)...
- Services: AttributeSequence Lookup(String name, AttributeType t), Bind(String name, AttributeSequence attr), UnBind(String name).
- Une composante d'un nom peut mener à un autre contexte. La recherche d'un nom peut donc se faire de manière itérative ou récursive en partant du contexte racine dont la localisation doit être bien connue.
- Cache: (nom, attributs), ou (Préfixe,nom de serveur).
- Différents espaces de noms: ordinateurs, usagers, services, fichiers...

Discussion

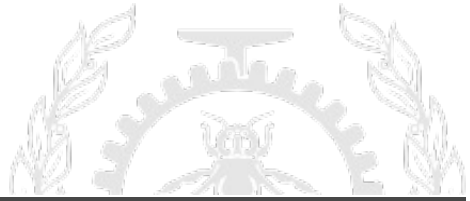
- Noms relatifs.
- Fusion d'espaces de noms.
- Alias pour aider restructuration.
- URI (*Uniform Resource Identifier*, peut être URL ou URN), URL (*Uniform Resource Locator*, comme un hyperlien) et URN (*Uniform Resource Name*, comme un ISBN par exemple).
- Le nom peut être un des attributs avec recherche possible sur tous les attributs: quel est le nom de l'usager dont le numéro de téléphone est X. (Pages jaunes au lieu de simples pages blanches).

Différents services

- Domain Name Service (DNS)
- Hesiod (Projet Athena au MIT)
- NIS (Sun Yellow Pages)
- Netinfo (NeXT)
- Banyan VINES
- NT Domains (avant Active Directory)
- X.500 (OSI)
- LDAP (simplification de X.500)
- Active Directory (Microsoft, basé sur LDAP)
- SLP, Jini, CORBA naming service, Portmap...

DNS

- Avant 1987, chacun prenait une copie d'un monstrueux fichier `/etc/hosts` par ftp.
- Convertir les noms en adresses IP.
- Trouver le serveur de courriel pour un domaine.
- Informations sur chaque ordinateur.
- Alias pour services courants (www.polymtl.ca, `ntp.polymtl.ca`, `ftp.polymtl.ca`).
- Trouver le nom pour une adresse IP.

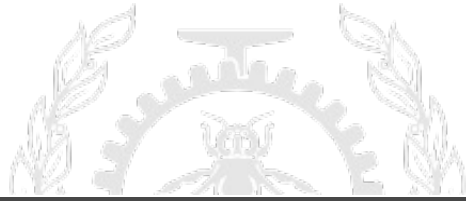


Organisation de DNS

- Serveur avec: attributs pour les noms d'un domaine, noms et adresses des serveurs en autorité pour le domaine et pour les sous-domaines dont l'autorité a été déléguée, paramètres pour la zone comme le TTL (Time To Live).
- Chaque zone doit être servie par au moins deux serveurs en autorité qui présentent des modes de défaillance non corellés.
- Le logiciel de serveur Bind peut être configuré en serveur primaire, secondaire ou cache seulement.
- La librairie client contacte par UDP les serveurs qui s'occupent de maintenir un cache en utilisant les valeurs de TTL.
- Peut avoir plusieurs IP (avec TTL très court) pour un nom, de manière à répartir des requêtes sur plusieurs ordinateurs.

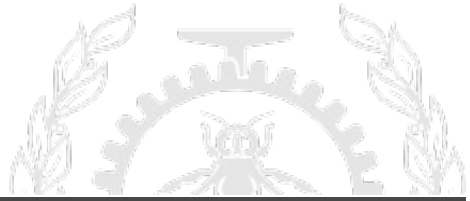
Serveurs DNS de départ

- Le service officiel est géré par le Internet Corporation for Assigned Names and Numbers (ICANN), anciennement sous le USA Department of Commerce et, depuis octobre 2016, après 18 ans de débats, sous une gouvernance plus neutre.
- OpenNIC offre des serveurs racine alternatifs avec beaucoup plus de liberté sur les noms de domaines disponibles (.bbs, .free, .geek, .libre, .neo, .null, .pirate. . .).
- Il existe d'autres racines DNS alternatives comme New Nations (.ko, .ku, .te, .ti, .uu. . .).



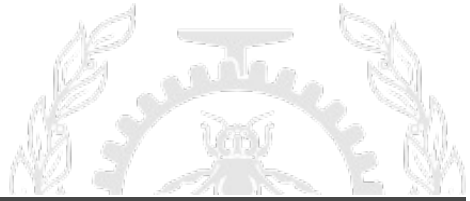
Jini / Apache River

- Développé par Sun sous le nom Jini, puis transféré à Apache sous le nom de projet River.
- Message à tous pour trouver le serveur.
- Enregistrement des services offerts par chaque objet avec un TTL.
- Requêtes pour découvrir les services appropriés.
- Appariement des requêtes basé sur la hiérarchie de types Java.



Service Location Protocol (SLP)

- Message à tous pour chercher un service (avec certains attributs).
- Les serveurs SLP qui connaissent un tel service répondent.
- SLP est souvent utilisé pour localiser des imprimantes et est supporté par des systèmes d'impression comme CUPS

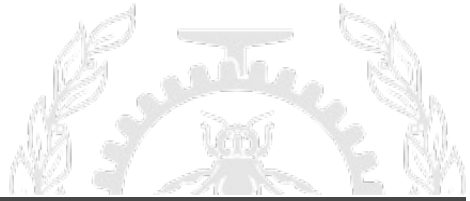


Le service de noms de OSI: X.500

- Wikipedia cite: *The original X.500 plan is unlikely ever to come to fruition*, mais LDAP s'en inspire.
- DIT (Directory Information Tree): hiérarchie de noms répartie sur plusieurs serveurs.
- DIB (Directory Information Base): noms et ensembles d'attributs pour chaque nom.
- Agent usager: accédé par les applications.
- Agent serveur: fournit les réponses, possiblement en faisant des requêtes à d'autres serveurs, ou en redirigeant le client vers un autre serveur.
- Les types des attributs sont définis avec ASN.1 (Abstract Syntax Notation). Un des attributs est le nom.
- Opérations: lire (chemin et liste des attributs désirés), chercher (préfixe de chemin, et expression booléenne de tests sur les valeurs des attributs).
- Interface d'administration pour la mise à jour.

LDAP

- Utilisé dans les grosses entreprises à la place de NIS
- Version allégée de X500 basée sur TCP/IP plutôt que OSI.
- Utilise ASN.1 et BER.
- Connexion TCP port 389.
- Arbre de répertoires contenant des entrées, chaque entrée constituée d'attributs pouvant contenir plusieurs valeurs.
- Requêtes et réponses asynchrones (plusieurs requêtes de suite, réponses non ordonnancées).



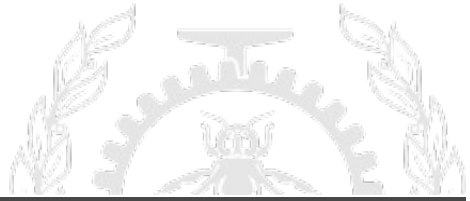
LDAP: opérations

- Start TLS: passer en mode encrypté.
- Bind: s'authentifier et spécifier la version du protocole.
- Search: effectuer une recherche dans le répertoire.
- Compare: vérifier si une entrée a une certaine valeur comme attribut.
- Add: ajouter une nouvelle entrée.
- Delete: effacer une entrée.
- Modify: modifier une entrée.
- Modify Distinguished Name (DN): renommer ou déplacer une entrée.
- Abandon: annuler une requête envoyée.
- Extended Operation: mécanisme pour extension.
- Unbind: fermer la connexion.

LDAP: recherche

- baseObject: chemin absolu de l'entrée à laquelle commencer la recherche.
- scope: entrée, répertoire ou sous-arbre à chercher.
- filter: critères de recherche, combinaisons (et ou non) sur des relations (égal, commence par...) sur les valeurs des attributs.

- derefAliases: suivre ou non les alias.
- attributes: quels attributs retourner dans les résultats.
- sizeLimit, timeLimit: temps maximum et taille maximum des résultats à retourner.
- typesOnly: seulement retourner le type des attributs et non leur valeur.

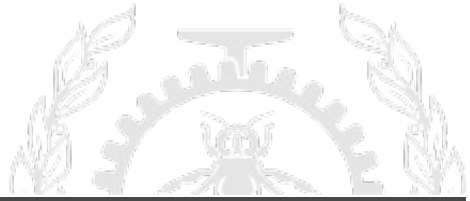


Organisations sans hiérarchie

- Les adresses Ethernet sont uniques et contiennent assez d'information pour identifier leur fabricant. Il n'y a aucune correspondance avec leur propriétaire ou le réseau sur lequel elles sont connectées.
- Les adresses IP sont très bien structurées à l'intérieur d'une organisation. L'adresse de réseau de chaque organisation est plutôt ad hoc.
- Les numéros de téléphone identifiaient le pays, le quartier et le domicile de manière hiérarchique. Avec la compétition pour les services de téléphonie, la possibilité de transférer son numéro, et la téléphonie mobile, ce n'est plus le cas.
- Pour les services offerts à très grande échelle, on veut assigner un identificateur unique à chaque objet (e.g. fichier). Un client doit pouvoir trouver l'objet sur un des nombreux serveurs sans avoir à faire une recherche sur un serveur central.

Recherche par envoi à tous

- Pour trouver un objet avec une certaine adresse, on envoie un message à tous les serveurs, et celui qui le contient répond.
- Le protocole ARP (Address Resolution Protocol) permet d'envoyer un message à tous sur le réseau Ethernet local, afin de trouver l'adresse Ethernet correspondant à l'adresse IP cherchée.
- La taille de chaque réseau local Ethernet est limitée.
- Un cache est maintenu dans chaque ordinateur connecté au réseau.
- Le commutateur qui relie les ordinateurs sur le réseau local peut apprendre les adresses Ethernet et filtrer les envois à tous.



Routage

- Une table dans un routeur peut mémoriser la localisation des différents objets.
- A l'intérieur d'une organisation comme Polytechnique, la passerelle vers chaque sous-réseau de l'institution change peu (e.g. la station l4712-01.info.polymtl.ca a une adresse de 132.207.12.33 qui commence par 132.207.12 et est accessible via la passerelle 132.207.12.1 sur le 5ème port du routeur principal).
- A l'extérieur des organisations, il y a peu de structure et les tables de routage peuvent devenir très complexes pour savoir sur quel port envoyer les paquets destinés à un des millions de réseaux IP.
- Un routeur dont les tables sont limitées peut décider d'envoyer les paquets de destination inconnue à un routeur mieux informé.

Base de donnée centrale pour le point d'attache

- Un dispositif mobile peut se retrouver n'importe où dans le monde, il faut le localiser sans effectuer un envoi à tous planétaire.
- Une base de donnée HLR (Home Location Register) contient la position courante pour chaque abonné de téléphonie mobile.
- A chaque minute, un téléphone mobile sonde les antennes disponibles autour de lui et s'associe à une spécifique.
- L'antenne d'attache du téléphone mobile est enregistrée dans sa base de donnée HLR.
- Lorsqu'un appel est destiné à un téléphone mobile, le signal est envoyé directement à son antenne courante d'attache.

Localisation sans serveur central

- Une fonction de hachage calcule directement l'identificateur du serveur, à partir de l'objet recherché, sans passer par un serveur central.
- La fonction de hachage doit répartir uniformément les objets entre les serveurs. En cas de débalancement, ou d'ajout ou retrait de serveur, la fonction de hachage peut être modifiée. Certains objets doivent être déplacés pour que leur position corresponde à la valeur donnée par la nouvelle fonction de hachage.
- Le défi est d'avoir une fonction de hachage qui répartit bien les objets et pour laquelle il est possible de faire des modifications qui ne changent pas beaucoup les assignations antérieures.
- Les services de fichiers (e.g. Ceph) et de tables de hachage distribuées (e.g. Cassandra, Chord) basés sur ce principe sont très populaires.

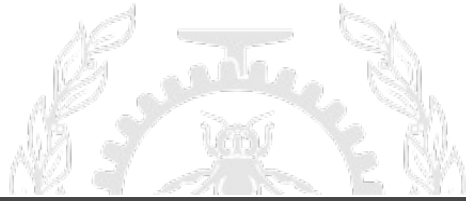
Discussion

- Le service DNS demeure le point d'entrée sur l'Internet. C'est un service facile à offrir mais essentiel. Il ne sert pratiquement que pour résoudre les adresses IP.
- Les autres services de répertoires (LDAP, Active Directory ou NIS) sont internes à une organisation et contiennent principalement la base de données des usagers (avec leur mot de passe, localisation du répertoire de fichiers, quotas. . .).
- Les requêtes externes passent souvent par une interface Web ou des Web Services (e.g. bottin du personnel de Polytechnique).
- Les données qui alimentent ces services peuvent souvent être stockées dans les faits dans une base de donnée conventionnelle.

Problèmes des Alias Ex. 7.1

Quels sont les problèmes associés aux alias dans un service de noms?

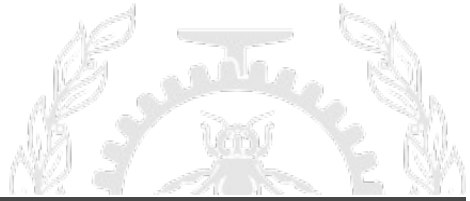
Les alias sont utiles lorsque plusieurs noms sont applicables à une ressource. Par contre, ils peuvent prêter à confusion puisque deux noms différents mènent à la même chose. Les alias au niveau des répertoires peuvent conduire à des cycles qui peuvent être des écueils pour les outils de navigation dans l'espace de noms.



Utilité d'un cache négatif - Ex. 7.2

Quel est le problème des noms inconnus dans un système par envoi à tous?

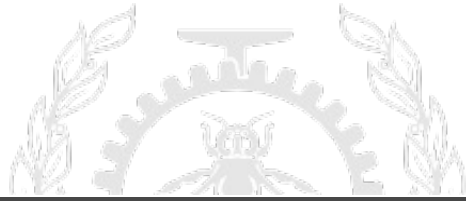
Un système par envoi à tous permet à plusieurs serveurs de collaborer facilement, pour des fins de redondance ou de division de l'espace de noms. Cependant, lorsqu'un nom inconnu est demandé, aucun serveur ne répond et le client ne sait pas si sa requête s'est perdue ou si le nom est inconnu des serveurs. Certains systèmes incluent des entrées négatives, l'information qu'un certain nom n'existe pas.



Robustesse par les caches - Ex. 7.3

Comment les caches peuvent-ils augmenter la disponibilité du service de noms?

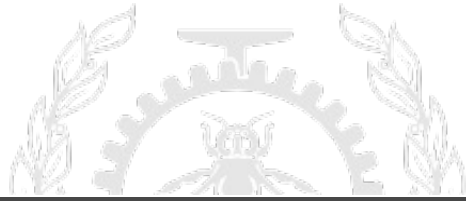
L'idéal est d'obtenir la précision et la simplicité d'un système centralisé et la robustesse et performance d'un système réparti. Les caches permettent d'obtenir un excellent compromis. Les caches ne requièrent aucune configuration, ils ne font que mémoriser pour leur période de validité (TTL) les informations qu'ils voient passer; ce faisant ils peuvent servir une grande proportion des requêtes.



Le point final - Ex. 7.4

Pourquoi ne pas mettre de point à la fin des noms d'ordinateurs (e.g. www.polymtl.ca. qui est comme /ca/polymtl/www)?

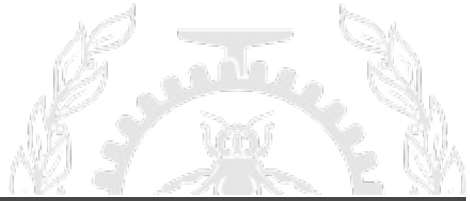
Les noms relatifs ne sont pas supportés dans les DNS, le seul cas particulier est le nom de la dernière composante seul. Dans ce cas, il n'y a pas d'ambiguïté puisqu'aucun nom absolu n'a moins de deux composantes.



Sauver un niveau - Ex. 7.5

Pourquoi les serveurs à la racine (e.g. .com, .org, .edu...) contiennent-ils l'information pour deux niveaux?

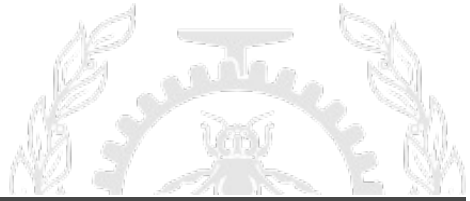
Les noms au premier niveau sont très peu nombreux (.com, .edu...) et ne correspondent pas à des entités différentes. Ils peuvent donc être servis ensemble et inclure le second niveau. Un serveur séparé s'occupe cependant des noms correspondant à chaque pays (e.g. serveur pour .ca).



Itération et récursion - Ex. 7.6

Quels sont les avantages respectifs des accès récursifs versus itératifs pour la résolution de noms?

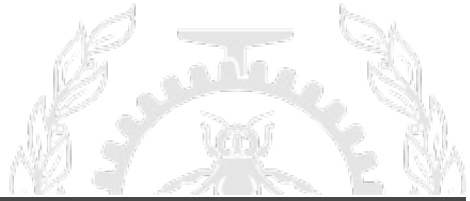
L'accès récursif est simple et permet au serveur de voir la réponse et de la conserver en cache. Par contre, s'il bloque pendant l'accès récursif, la performance en souffrira pour les accès concurrents. Lors de l'accès itératif, le client est référé à un autre serveur et le serveur peut facilement passer à la prochaine requête.



Réponses multiples - Ex. 7.7

Quand un serveur de noms fournit-il plusieurs réponses à une requête?

Un domaine peut avoir plusieurs serveurs de noms ou de courrier. Ils sont tous retournés lors d'une requête pour ces services.



Sécurité - Ex. 7.8

Quelles mesures de sécurité doivent entourer un service de nom?

La modification/ajout des entrée ne doit être possible que par les administrateurs, sauf pour certains champs (adresse, mot de passe) qui peuvent être réservés à l'utilisateur.

La délégation de portions de l'espace de nom doit être possible, et la consultation de certaines informations peut être restreinte. Il faut bien sûr que l'usager puisse être certain de parler au serveur de nom et que le serveur de nom puisse aussi authentifier l'usager.