Le Niveau Application

De l'Internet

G. Florin

Conservatoire National Des Arts Et Métiers

Plan du cours

Internet: Généralités

L’Internet s'est développé a partir des travaux sur le réseau ARPANET puis sur l'interconnexion de réseaux aboutissant au protocole IP.

- Un ensemble de liaisons physiques et de routeurs entre des réseaux locaux, régionaux.

- Utilisant les protocoles normalisés aux niveaux liaison, réseau et transport (TCP/IP).

L’Internet doit son succès au développement d’un ensemble d'applications normalisées

- Messagerie, transfert de fichiers, systèmes d’informations, annuaires ...

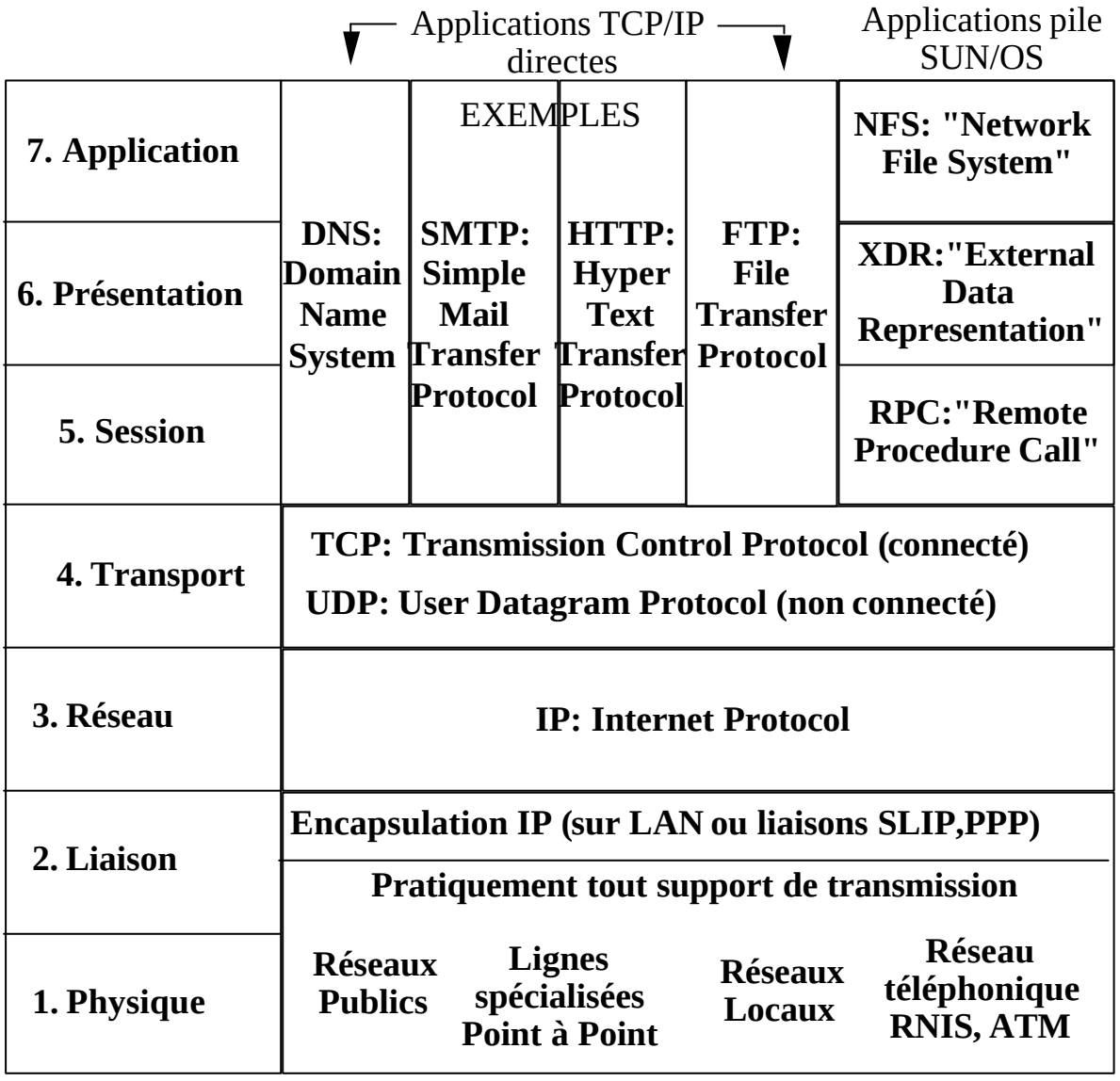
L’Internet est soumis à un contrôle international de plus en plus rigoureux.

Les protocoles d'application

L'Internet est essentiellement connu maintenant pour son ensemble de **protocoles d'applications** normalisés très utilisés.

- Pour **réaliser** des fonctions de nommage:
  - Serveur de nom: dns
- Pour **accéder** à des machines distantes
  - Accès interactif à distance : telnet
- Pour **communiquer** entre personnes
  - Messagerie : smtp
- Pour **transférer** des fichiers
  - Transfert de fichiers : ftp
- Pour **échanger** à des informations
  - Client-serveur d'accès aux informations: web
- Pour **administrer** le réseau
  - Administration de réseaux : snmp

La pile des protocoles Internet



Quelques Dates

1966 - 1969 : Préparation du projet Arpanet (réflexions, appel d'offre construction des commutateurs).
1969 : Arpanet fonctionne entre 4 commutateurs.
1971 : 15 commutateurs, 24 hôtes
1972 : Spécification du premier accès distant
1973 : Spécification du premier transfert de fichiers
1974 : Première utilisation du terme **Internet**.
1977 : RFC messagerie électronique **->** **smtp**.
1978 : Travaux sur TCP/IP séparation TCP et IP
1982 : Normalisation complète de TCP IP.
1983 : Séparation Milnet. et Arpanet **->** Internet
1984 : Première version de **dns**.
1986 : Apparition des **news**.
1991 : Apparition du **web (html, http)** (CERN Genève)
2000 : Les réseaux se comptent en centaines de milliers et les hôtes en centaines de millions.

Les organes de contrôle de l'Internet (1)

ISOC "Internet SOCiety"

L'organisation **principale** des professionnels de l'Internet : chargée de la croissance, de l'évolution technique, des aspects sociaux, politiques (comporte des chapitres régionaux.

IAB "Internet Architecture Board"

Définition de **l'architecture** du réseau, de ses protocoles, arbitrage des conflits entre différentes tendances.

IESB "Internet Engineering Steering Group"

Adopte les normes sur proposition IETF**.**

IETF Internet Engineering Task Force

Définition, expérimentation des protocoles : groupes de travail par domaines, production des RFC "Request For Comments".

Les organes de contrôle de l'Internet (2)

IRTF "Internet Research Task Group"

Recherche à long terme.

IANA "Internet Assigned Number Authority" transformé en ICANN 'Internet Corporation for Assigned Names and Numbers'

Chargé de l'affectation des **adresses**, **noms** (délégation de certains espaces de nommage), **valeurs de paramètres**, ... pour l'ensemble des protocoles Internet.

Exemple: gestion des adresses **IP (**numériques), gestion des **noms** de domaines, affectation des **paramètres de protocoles**, des numéros de **ports bien connus**, gestion des **serveurs racines du DNS**, définition des **MIB** ("Management Information Base")**.**

I La gestion d'annuaires répartis

Le système des noms de domaines de l'Internet
DNS 'Domain Name System'

Introduction Historique
1 L'espace des noms de domaine
2 La base de données distribuée du DNS
3 Les serveurs DNS et la résolution des requêtes
4 Approfondissements
5 Exemples
Conclusion

Introduction

Contrainte de base très importante: les usagers préfèrent utiliser les noms logiques

Exemples

Adresse courrier : **gerard@cnam.fr**
Plutôt que : gerard@[163.173.128.60]
Nom de site web : **http://www.cnam.fr**
Plutôt que : **http://163.173.128.28**

- Besoin d'un ensemble de mécanismes:
  de création,
  d'administration,
  de mise en relation
pour des noms logiques, des adresses, des attributs.

- Besoin principal: association entre des noms logiques et des adresses physiques IP.

- Extension: associer à des noms logiques des attributs selon les besoins.

Illustrations

- Un **hôte** du réseau (ou un **service** accessible en réseau) reçoit un nom :

Exemple: fermi, bose, www, ftp …

- Ce nom est défini dans le contexte d'une **organisation identifiée** CNAM en France:

Exemple: fermi.cnam.fr, www.cnam.fr

- Un problème de base consiste à associer à un **nom logique** de l'hôte à son adresse IP :

Exemple : bose.cnam.fr : 163.173.128.28

- On peut réaliser à partir de là d'autres **correspondances :**

Exemple : Notion d'alias
www.cnam.fr <=> bose.cnam.fr

Historique : définition locale des correspondances noms - adresses

- Au début de l'Internet les noms étaient définis **localement** sur chaque hôte dans un **fichier** (Exemple: **/etc/hosts** en UNIX):
  adresse IP hôte (@IP) ↔ noms logique hôte

Exemple: début de fichier /etc/hosts

```
127.0.0.1        localhost
163.173.212.2    cisco-for-acces35
163.173.215.0    colombus.cnam.fr
163.173.215.2    aryaman
      …….
```

- /etc/hosts était mis à jour par ftp de nuit **automatiquement** ou **manuellement** à partir d'une version la plus à jour possible pour suivre l'évolution du réseau Internet.

- Internet est devenu trop **grand**, trop **évolutif**
  . **Consommation** de bande passante.
  . **Charge** de la machine serveuse.
  . **Délais** de mise à jour du fichier.
  . **Besoin** de services plus généraux.

Solution avec annuaires distribués: le DNS RFC 1034, 1035

Objectifs généraux du système de désignation

- Création d'un service **d'annuaire distribué** (base de données distribuée d'informations)

- Accessible au moyen **d'un espace de nommage hiérarchique unifié**.

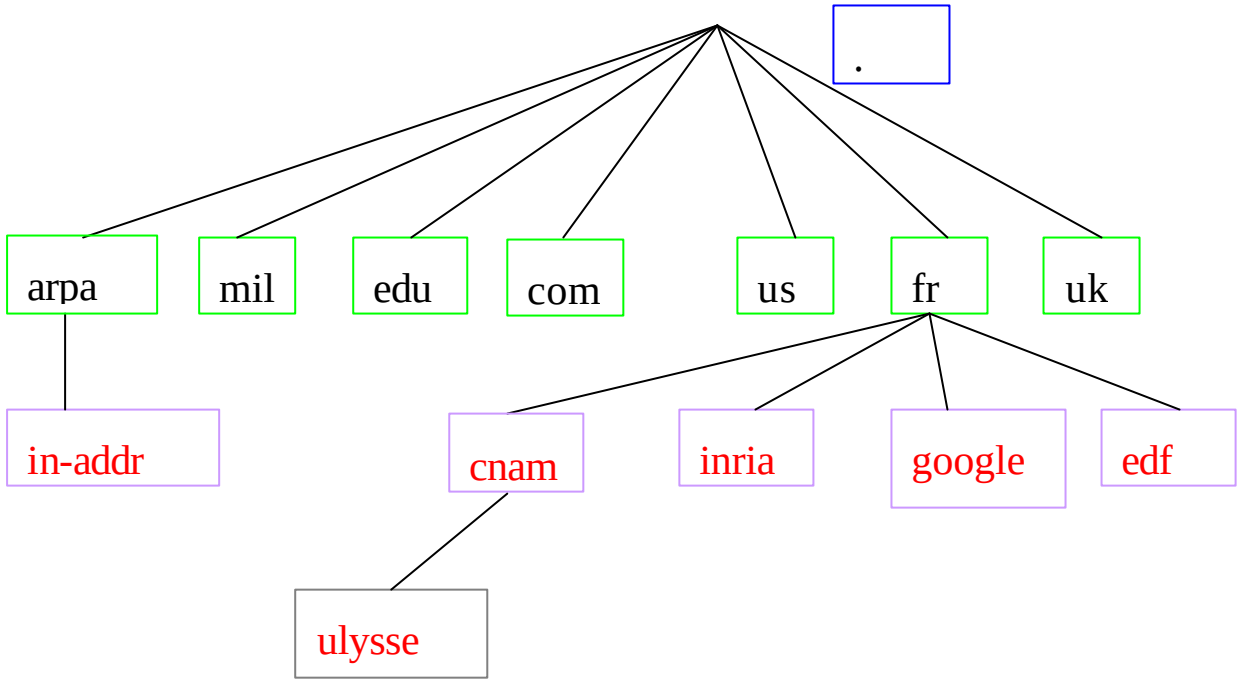
- **Fonction principale**: correspondance:
  nom logique de site <-> adresse IP.
  + Différentes autres fonctions d'accès.

- Le protocole client-serveur d'accès est utilisable par **plusieurs piles de protocoles**.

- Utilisable par tout système (petit ou grand)
  . une procédure client (le **résolveur**)
  . et un **serveur** DNS qui répond.

Un outil fondamental de l'Internet

1) L'espace des noms de domaines : « Domain Name Space » Une hiérarchie de noms logiques



Au niveau le plus haut: Plusieurs centaines de noms de domaines.

a) com : Noms génériques de domaines.
b) fr : Noms géographiques de domaines.
c) arpa : Corresp adresses IP vers noms.

Aux niveaux intermédiaires: Des noms de domaines (qui sont des sous-domaines).

Au niveau des feuilles: Des sous-domaines composés d'hôtes ou définissant des services.

Les domaines de plus haut niveau (TLD 'Top Level Domains')

Domaines génériques (organisationnels)

.com : Organismes commerciaux (Verisign)
.net : Prestataires réseaux (Verisign)
.aero : Industries aéronautiques (SITA)
.biz : Affaires (NeuLevel, Inc).
.coop : Associations cooperative (Dot Cooperation LLC).
.info : Orgs d'information (Afilias Limited)
.museum : Musées (Museum Domain Management Association).
.name : Individus (Global Name Registry).
.org : Autres organisations (Verisign)

Domaines non ouverts

.edu : Institutions d'éducation US
.gov : Organisations gouvernementales US
.mil : Armée US
.int : Organisations internationales.

Domaines Nationaux (géographiques)

Codes de pays ISO 3166

.fr : France
.uk : Grande-Bretagne
…..

.ac – Ascension Island
.ad – Andorra
.ae – United Arab Emirates
.af – Afghanistan
.ag – Antigua and Barbuda
.ai – Anguilla
.al – Albania
.am – Armenia
…..

Le domaine d'infrastructure ARPA

Pour l'obtention à partir des adresses IP les noms logiques de domaines associés.

La désignation arborescente (suite)

- Les feuilles de l'arbre DNS sont des noms **d'hôtes** ('ulysse', 'savitri') ou **de services** ('www', 'ftp').

- Les nœuds intermédiaires sont des noms de domaines composés **d'ensembles de ressources.**

- L'organisation arborescente du nommage permet de déléguer **l'administration** des noms.

- Pour créer un nom à l'intérieur d'un domaine il **suffit d'avoir l'autorisation** de ce domaine (de son administrateur).

- **L'arbre de désignation du DNS** n'a pas de relation avec **la topologie physique** du réseau (l'internet).

La structure des noms (1)

Les noms absolus ‘FQDN’ ‘Fully Qualified Domain Name’

- Ils sont construits en suivant l’arbre **des feuilles vers la racine** qui est notée ‘.’

- On sépare les noms intermédiaires par des .

- Exemple de FQDN : cnam.fr..

  . Le domaine **cnam** appartient au domaine **fr** qui appartient à la **racine** : séparation des chaînes associées aux domaines par des points (approche différente des noms de fichiers. On noterait /fr/cnam).

- **Notation des noms assez souple** :
  Pour faciliter l’usage: cnam.fr. ⇔ cnam.fr.. (suppression possible du point racine)
  On peut même utiliser : cnam.fr (qui est un nom relatif voir plus loin).

La structure des noms (2)

Les noms relatifs

- Ils n'ont **pas de point final** et doivent être **complétés par une chaîne** de caractère pour former un nom absolu utilisable.

La chaîne de caractère à ajouter doit être définie (souvent le nom du domaine courant).

- Exemple: recherche du nom **cs.vu.nl** dans le contexte cnam.fr..

on essaye **cs.vu.nl.cnam.fr..**=> échec

mais aussi **cs.vu.nl..** => succès

Compléments sur les noms de domaines

- Un nom de domaine a **moins de 63 cars**.

- Un nom complet a **moins de 255 cars**.

- La casse est **non significative**:

cnam = Cnam = CNAM.

2) La base de données distribuées du DNS

Les enregistrements ressources (1)

. Le DNS gère une **base de données répartie**
. D'unités d'informations baptisées **RR**
'Resources Records'
. Pour satisfaire des besoins **variés** de stockage et des protocoles différents.

Structure de donnée d'un RR

RR = {nom_de_domaine, durée_de_vie, classe, type, valeur}

- Le **nom de domaine** ('**name**') identifie un nœud de l'arborescence.

- La **durée de vie** (**TTL 'Time To Live'**) définit la durée de validité de l'information dans un cache (nombre entier de secondes).

Les enregistrements ressources (2)

- La **classe** (‘**class**’) identifie le protocole utilisateur (essentiellement **In** pour Internet).

- Le **type** (‘**type**’) identifie le **type de donnée** stockée : type adresse A, type serveur de noms d’un domaine NS, type relais de courrier MX.

- La longueur de la valeur (‘**rdlenght**’).

- La **valeur** (‘**rdata’**) contient les données significatives associées au type : une adresse, un nom de domaine, une chaîne de caractères.

Détails concernant les types (1)

SOA (‘Start Of Authority’) : Informations générales d’un domaine (serveur primaire, mail administrateur, numéro de série des infos utilisées, durée entre deux mises à jour, délai entre deux tentatives de mise à jour d’un secondaire sur si échec, durée de vie minimale des infos …).

{cnam.fr, 43200 (12 heures), IN, SOA,
origin = asimov.cnam.fr
mail addr = hostmaster.cnam.fr
serial = 2000031515
refresh = 21600 (6 heures)
retry = 3600 (1 heure)
expire = 1814400 (21 jours) }

A (‘Address’) : l’adresse d’un hôte (@IP).
{ulysse, 86400, IN, A, 163.173.136.6}

NS (‘Name Server’) : spécifie le nom d’un hôte serveur DNS pour un domaine.
{cnam.fr, 86400, IN, NS, asimov.cnam.fr}

Détails concernant les principaux types (2)

MX (‘Mail Exchanger’) : serveur de courrier électronique d’un domaine (avec une priorité)
{cnam.fr, 86400, IN, MX, 10, fermi.cnam.fr}

HINFO (‘Hardware Informations) : infos sur le matériel et le logiciel d’un hôte.
{bose.cnam.fr, 43200, IN, HINFO, sun unix}

CNAME (‘Canonical Name’) :un alias
{ftp.cnam.fr,86400,IN,CNAME,bose.cnam.fr}

TXT (‘Text’) : permet d’associer n’importe quel texte à un nom de domaine.
{cnam.fr, 43200, IN, TXT, Conservatoire
National des Arts et Métiers}

PTR (‘Pointer’) : un pointeur sur un autre nom (pour associer à une adresse IP un nom).
{6.136.173.163.in-addr.arpa, 86400, IN,
PTR, ulysse.cnam.fr}

Liste complète des types d'enregistrements ('Resource records')

Symbole (Code)	Signification	RFC
A (1)	Address.	(RFC 1035).
AAAA(28)	IPv6 address.	(RFC 1886).
AFSDB	AFS Data Base locat.	(RFC 1183).
CNAME(5)	Canonical NAME.	(RFC 1035).
HINFO(13)	Host INFOrmation.	(RFC 1035).
ISDN	ISDN.	(RFC 1183).
KEY	Public KEY.	(RFC 2065).
KX	Key eXchanger.	(RFC 2230).
LOC	LOCation.	(RFC 1876).
MB	MailBox.	(RFC 1035).
MG		(RFC 1035).
MINFO		(RFC 1035).
MR		(RFC 1035).
MX (15)	Mail eXchanger.	(RFC 1035).
NULL		(RFC 1035).
NS (2)	Name Server.	(RFC 1035).
NSAP	Network Service Access Point Redéfini par le RFC 1706.	

Liste des types d'enregistrements (2) ('Resource records')

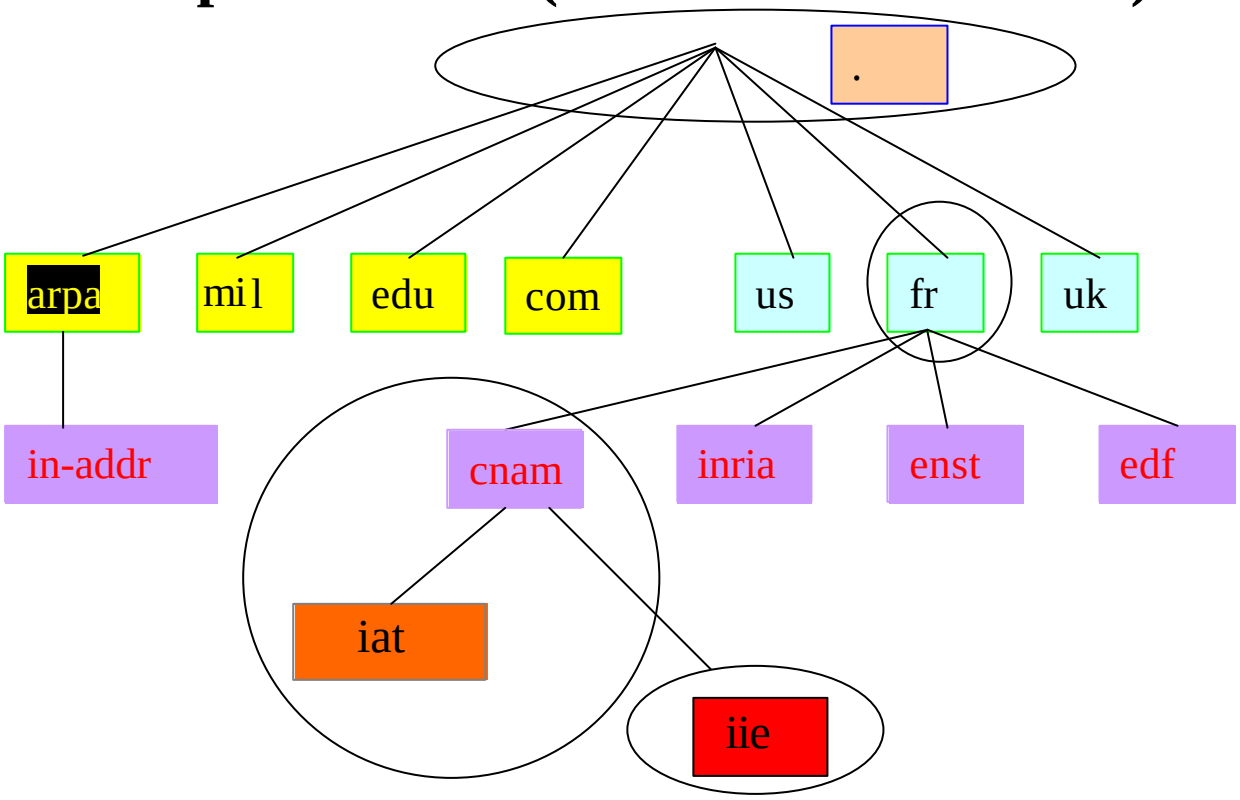
NXT		NeXT.	(RFC 2065).
PTR	(12)	PoinTeR.	(RFC 1035).
PX		Pointer X400/RFC822	(RFC 1664).
RP		Responsible Person.	(RFC 1183).
RT		Route Through.	(RFC 1183).
SIG		Crypto SIGnature.	(RFC 2065).
SOA	(6)	Start Of Authority.	(RFC 1035).
SRV		SeRVer.	(RFC 2052).
TXT		TeXT.	(RFC 1035).
WKS		Well-Known Service.	(RFC 1035).
X25		X25.	(RFC 1183).

3) Les serveurs DNS et la résolution des requêtes

Notion de zone

- Une **zone** est une **unité d'administration** (tous les membres d'une zone sont servies par un même serveur).
- Une zone regroupe **un ensemble de domaines voisins** qui ne se recouvrent pas.

Exemple de zones (associées à des ovales)



Différences entre Domaine et Zone

- Un domaine définit un ensemble de noms qui ont un même suffixe => c'est **un découpage syntaxique** de l'espace de nommage Internet.

- Une zone est un ensemble de noms ayant un même suffixe et servis par le même serveur de nom => c'est **un découpage administratif** définissant **la portée d'action** des serveurs de noms.

. Un serveur de nom pour une zone peut **servir différents sous domaines**.

Exemples (schéma précédent)

- La zone **CNAM** possède des **serveurs de nom** qui servent aussi le sous-domaine **iat.**

- Le serveur CNAM ne traite pas le service du sous domaine **iie** qui à un **serveur de zone en propre.**

Notion de serveur primaire

- Une zone est servie par un (ou plusieurs) serveurs **primaires** alimentés directement en informations par l'administrateur système.

- Les fichiers d'informations des serveurs **primaires 'font autorité'** ('Authoritative Answer').

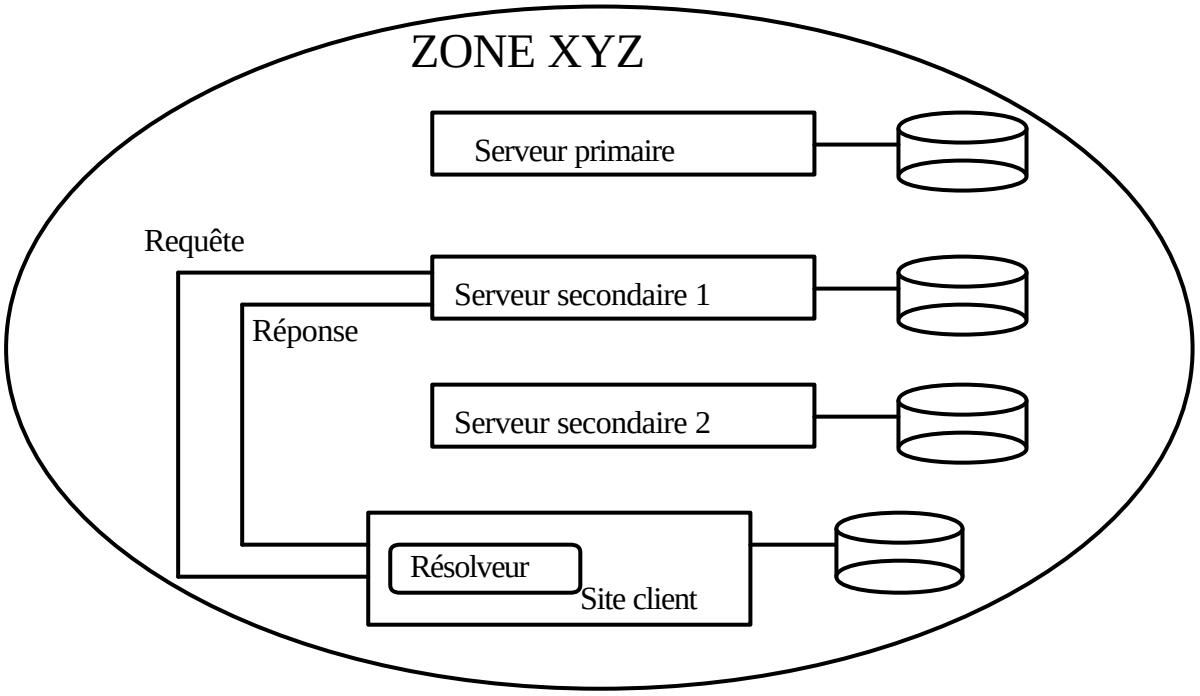
Notion de serveur secondaire

- Pour des raisons de **performances** (partage de charge) et de **fiabilité** (tolérance aux pannes de serveurs) on créé des serveurs secondaires.

- Les serveurs secondaires ne sont pas alimentés directement mais **recopient périodiquement** en utilisant TCP la base d'informations DNS d'un serveur primaire.

Notion de client DNS

- On appelle résolveur une procédure (exemple ***gethostbyname ou gethostbyaddr)*** invoquée sur un client pour un accès DNS.



Notion de cache

- Les serveurs primaires ou secondaires conservent dans un cache les réponses à des requêtes récentes :
  - . pour **gagner du temps**.
  - . pour éviter de **charger le réseau**.

Fonctionnement des caches

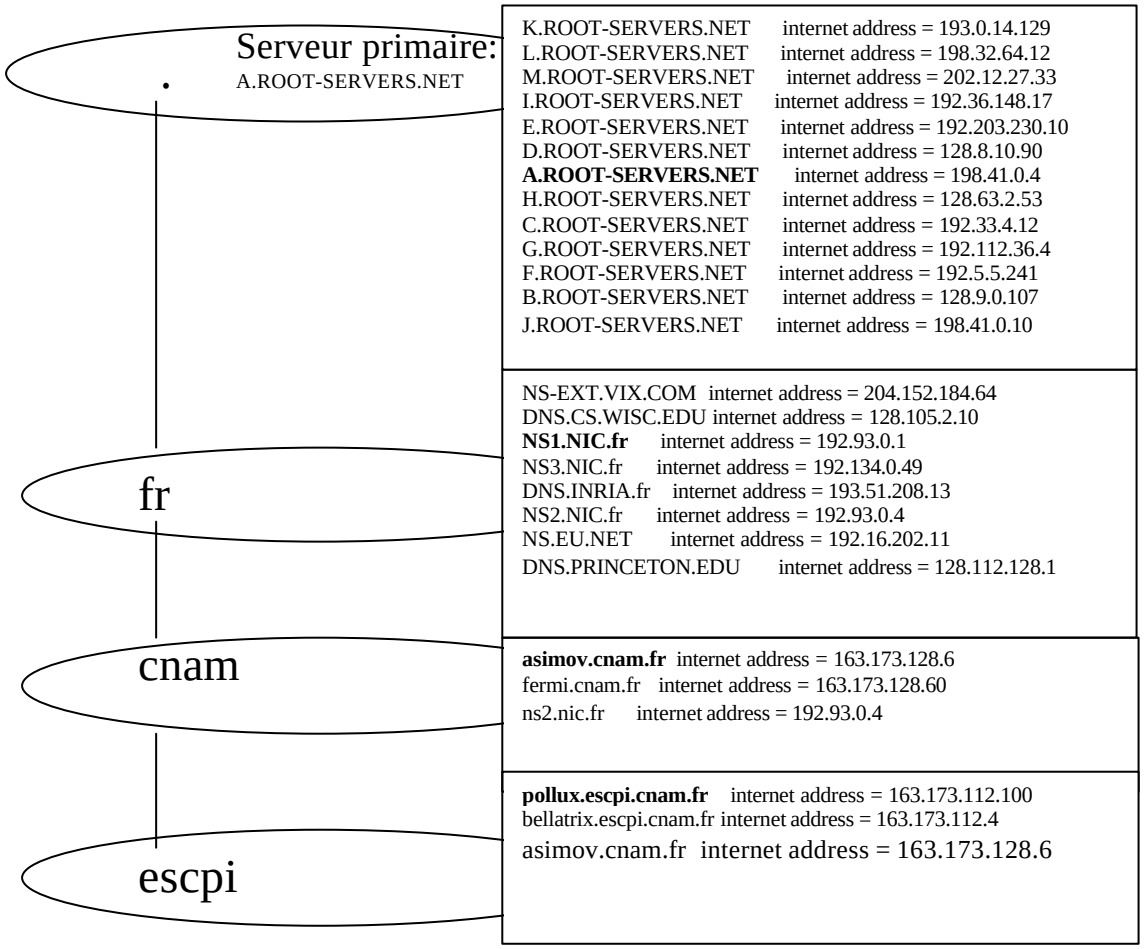
- Les serveurs DNS **gèrent des caches** (certains serveurs ‘proxy’ sans base de données gérènt **uniquement un cache**).

- **Problème de la cohérence des caches**: que se passe t’il quand l’administrateur modifie le serveur primaire.

- **Solution DNS** : **cohérence très faible** Simple limitation de **la durée de vie** dans un cache (notion de TTL ‘Time To Live’).
  .Les données stockées **changent lentement** (cohérence faible suffisante).
  . On donne la **priorité à la rapidité d’**accès sur la garantie de cohérence.

- **Information de cohérence**: Réponse d’un serveur primaire: ‘**Authoritative answer**’, réponse d’un secondaire ou d'un cache: ‘**Non-authoritative answer**’.

Exemple : zones et serveurs associés



Remarque : Un serveur secondaire de zone peut être localisé dans une autre zone.

La résolution des requêtes

- La résolution d'une requête est fondée sur une recherche qui peut **concerner plusieurs serveurs avant de trouver la réponse**.

- Le résolveur s'adresse tout d'abord à un **serveur DNS local** (un serveur primaire ou secondaire du domaine courant) **en présentant une question** concernant un nom de domaine.

. Si le serveur **local connaît la réponse** (dans sa base de données ou dans son cache) retour au client de la réponse.

. Si la requête porte sur un sous domaine : **Recherche dans les sous domaines.**

. Sinon **recherche sur d'autres serveurs DNS** qui peuvent être parcourus selon deux méthodes :

Recherche **récursive**.
Recherche **itérative**.

La recherche itérative

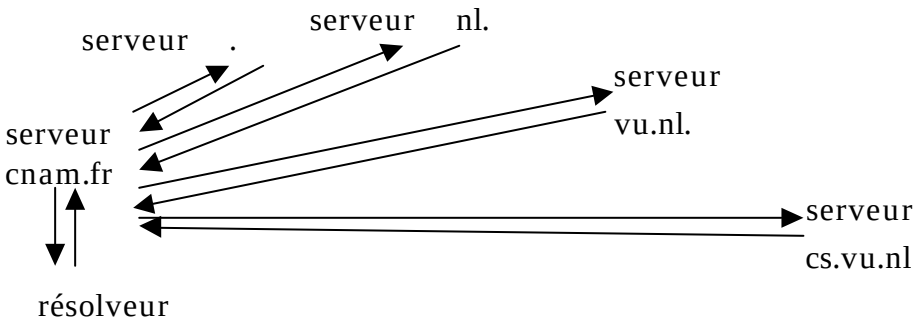
- On **interroge successivement** les serveurs.

- **Exemple :** Au cnam pour trouver le nom star.cs.vu.nl.. le résolveur interroge :

. Le serveur local **cnam.fr**

. Le serveur local interroge le serveur **racine** du DNS, puis le serveur de **nl**, puis celui de **vu.nl**, puis celui de **cs.vu.nl**

. Chaque serveur visité donne **l'adresse IP du serveur suivant** à interroger.



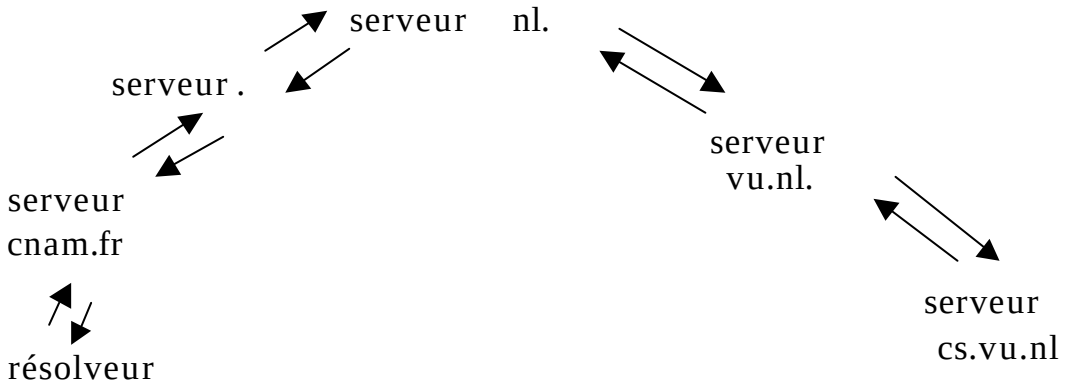
La résolution récursive

- Un indicateur dans la requête décrit la façon de la traiter : **itérative ou récursive.**

- Chaque serveur visité prend l'initiative **d'interroger le serveur suivant** pour obtenir pour lui même la réponse à la question posée.

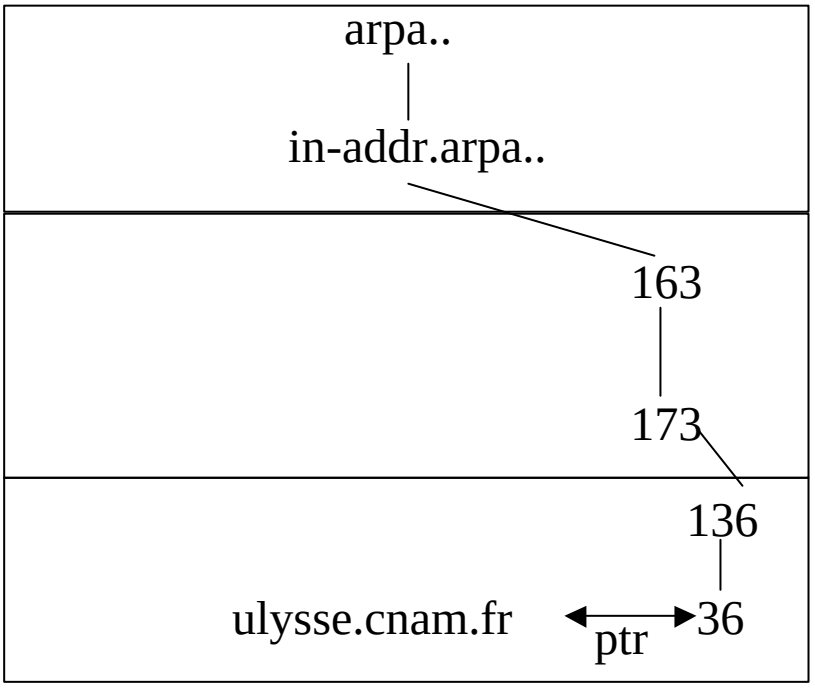
- La réponse revient en **visitant tous les sites**.

- On note la résolution effectuée dans les **caches** de tous les serveurs visités.



Les requêtes inverses

- Une requête inverse est **une requête formulée à partir de la valeur d'un attribut** et non d'un nom symbolique de domaine (d'un clé d'accès).
- **Principal besoin**: obtenir un nom de domaine à partir d'une adresse IP.
- Utilisation du **domaine spécial arpa, sous domaine in-addr** : structure logique du domaine.



Question: {IN, PTR, 36.136.173.163.in-addr.arpa, ?}
Réponse:36.136.173.163.in-addr.arpa name = ulysse.cnam.fr

4) DNS : Approfondissements

4.1 DNS : Le protocole

Un format pour les requêtes et les réponses

.

0 15

Identifiant
Drapeaux
Nombre de questions ‘qdcount’
Nombre de réponses ‘ancount’
Nombre serveurs faisant autorité ‘nscount’
Nombre adresses serveurs arcount
Zone des questions (longueur variable)
Zone des réponses (longueur variable)
Zone des serveurs faisant autorité (longueur variable)
Zone des informations additionnelles Adresses des serveurs (variable)

Les différent champs (1)

Identification: Estampille 16 bits positionnée par le client pour associer une requête et une réponse (les serveurs DNS sont sans état).

Drapeaux : Une série d'indicateurs

1	4	1	1	1	1	3	4
QR	Opcode	AA	TC	RD	RA	Zéro	Rcode

QR : question (code 0) réponse (code 1)
Opcode : code opération (3 à 15 inutilisés)
0 question standard (sur un nom).
1 question inverse sur adresse (obsolete).
2 demande de status serveur.
AA : code réponse faisant autorité d'un primaire (Authoritative Answer) ou non.
TC : réponse tronquée à 512 octets (en UDP).
RD : demande de résolution récursive.
RA : réponse obtenue récursivement ou non.
Rcode :0 réponse correcte.
1 format d'interrogation incorrecte.
2 erreur serveur.
3 erreur nom de domaine (inexistant).

Les différent champs (2)

Qdcount: Nombre de champs questions dans un message (16 bits).

Ancount: Nombre de champs réponses (16 bits) obtenues directement.

Nscount: Nombre de zones d'adresses de serveurs primaires ou secondaires qui peuvent répondre (RR de serveurs primaires et secondaires pouvant donner des réponses faisant autorité).

Qdcount: Nombre de champs utiles supplémentaires.

Les champs ‘question’

0 7 8 15
Zone nom logique ‘qname’
Zone classe ‘qclass’
Zone type ‘qtype’

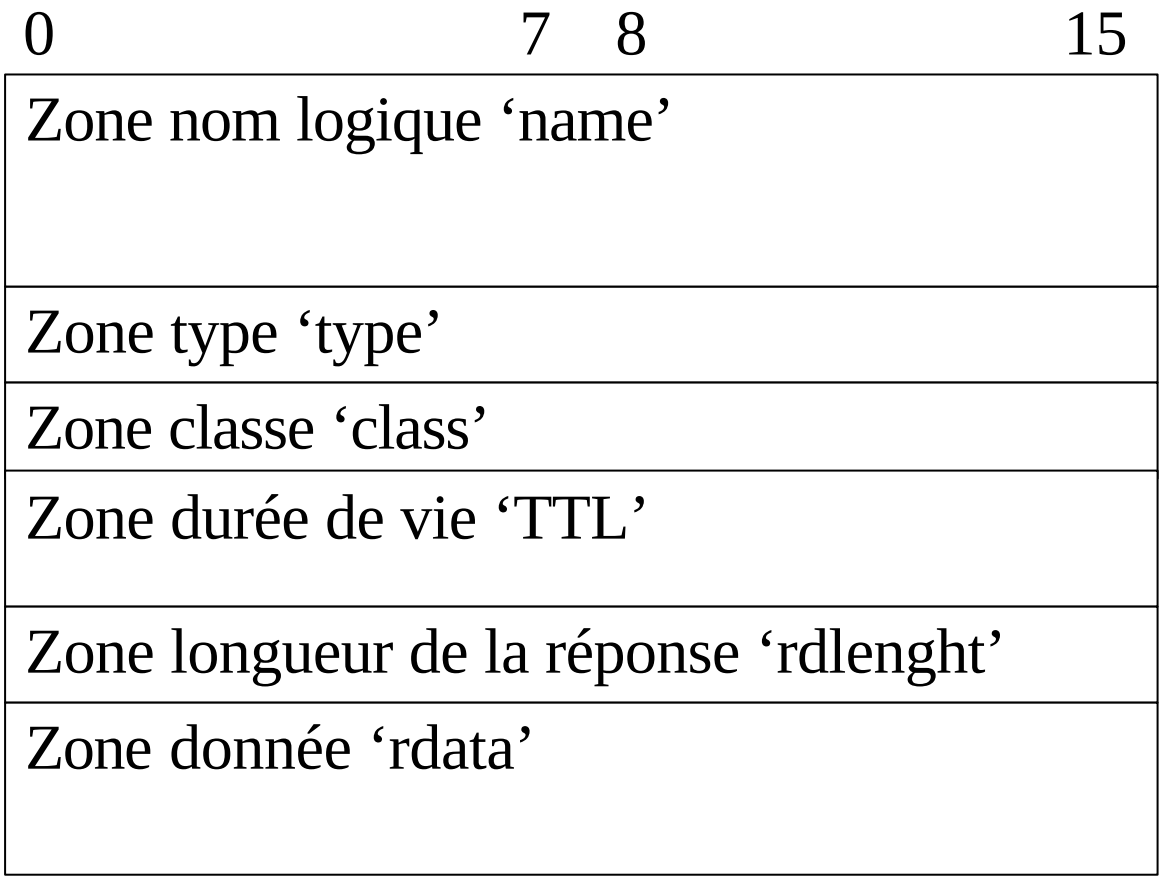
Qname: La chaîne de caractère associée au nom logique recherché. Chaque caractère est codé en ASCII et le nombre de caractères de chaque sous chaine est donné avant la chaîne.

04 63 6E 61 6d 02 66 72 00
c n a m f r

Qclass: Le type du réseau (1 pour Internet).

Qtype: Le type d’information demandée par la question (voir codes numériques des types).

Les champs ‘réponse’ (Resource Record)



Name: La chaîne de caractère associée au nom logique recherché.
Type: Le type de la réponse.
Class: Le réseau concerné (1 pour Internet).
TTL: La durée de vie dans un cache.
Rdlenght: La longueur de la zone réponse (exemple @IP 4 octets, @IPV6 16 octets …).
Rdata : Valeur de la zone réponse selon type.

Quelques caractéristiques complémentaires

- Une implantation du DNS très répandue : **BIND** (‘**B**erkeley **I**nternet **N**ame **D**omain’).

- Utilisation des **protocoles UDP ou TCP** numéro de port bien connu 53.

. En fait DNS **utilise au départ UDP** (pour aller vite) sur le numéro de port 53.

. Si la réponse a été **tronquée à 512 octets** en UDP (TC bits à 1) **réémission de la requête en TCP** sur le numéro de port 53 pour obtenir toute la réponse.

. **Autre utilisation de TCP** : pour transférer périodiquement les informations relatives à une zone entre serveurs primaires et secondaires.

5) Exemples d'utilisation du DNS
La commande nslookup

Il existe d'autres outils clients DNS plus ou moins comparables comme **host** ou **digger.**

Lancement

```
$ nslookup
Default Server:  asimov.cnam.fr
Address:  163.173.128.6
```

Recherche d'adresse IP et de nom

```
> set type=A
> ulysse.cnam.fr
Server:  asimov.cnam.fr
Address:  163.173.128.6

Name:    ulysse.cnam.fr
Address:  163.173.136.36
>
```

Recherche inverse d’un nom par l'adresse

```
>       set type=A
>       163.173.136.36
Server:  asimov.cnam.fr
Address:  163.173.128.6

Name:    ulysse.cnam.fr
Address:  163.173.136.36
>       36.136.173.163.in-addr.arpa..
Server:  asimov.cnam.fr
Address:  163.173.128.6

*** asimov.cnam.fr can't find 36.136.173.163.in-addr.arpa:
                                                Non-existent host/domain
>       set type=ptr
>       36.136.173.163.in-addr.arpa..
Server:  asimov.cnam.fr
Address:  163.173.128.6

36.136.173.163.in-addr.arpa        name = ulysse.cnam.fr

173.163.IN-ADDR.ARPA    nameserver = asimov.cnam.fr
173.163.IN-ADDR.ARPA    nameserver = fermi.cnam.fr
173.163.IN-ADDR.ARPA    nameserver = ns2.nic.fr
asimov.cnam.fr  internet address = 163.173.128.6
fermi.cnam.fr   internet address = 163.173.128.60
ns2.nic.fr    internet address = 192.93.0.4
```

Informations générales d'un domaine

```
> set type=SOA
> escpi.cnam.fr
Server:  asimov.cnam.fr
Address:  163.173.128.6

escpi.cnam.fr
        origin = pollux.escpi.cnam.fr
        mail addr = Postmaster.escpi.cnam.fr
        serial = 2000011016
        refresh = 21600 (6 hours)
        retry   = 3600 (1 hour)
        expire  = 1209600 (14 days)
        minimum ttl = 86400 (1 day)
escpi.cnam.fr   nameserver = pollux.escpi.cnam.fr
escpi.cnam.fr   nameserver = bellatrix.escpi.cnam.fr
escpi.cnam.fr   nameserver = asimov.cnam.fr
pollux.escpi.cnam.fr    internet address = 163.173.112.100
bellatrix.escpi.cnam.fr internet address = 163.173.112.4
asimov.cnam.fr  internet address = 163.173.128.6
>
```

Recherche de serveur de courrier

```
> set type=MX
> escpi.cnam.fr
Server:  asimov.cnam.fr
Address:  163.173.128.6

escpi.cnam.fr preference = 20, mail exchanger =
                                                fermi.cnam.fr
escpi.cnam.fr preference = 10, mail exchanger =
                                            postman.escpi.cnam.f
                                        r
escpi.cnam.fr   nameserver = pollux.escpi.cnam.fr
escpi.cnam.fr   nameserver = bellatrix.escpi.cnam.fr
escpi.cnam.fr   nameserver = asimov.cnam.fr
fermi.cnam.fr   internet address = 163.173.128.60
postman.escpi.cnam.fr   internet address = 163.173.112.5
pollux.escpi.cnam.fr    internet address = 163.173.112.100
bellatrix.escpi.cnam.fr internet address = 163.173.112.4
asimov.cnam.fr  internet address = 163.173.128.6
>
```

Recherche de toutes les informations

```
> set q=any
> escpi
Server:  asimov.cnam.fr
Address:  163.173.128.6
escpi.cnam.fr   internet address = 163.173.112.5
escpi.cnam.fr   preference = 10,
                          mail exchanger = postman.escpi.cnam.fr
escpi.cnam.fr   preference = 20, mail exchanger = fermi.cnam.fr
escpi.cnam.fr   text = "Unauthorized access to this network is
                                                              prohibited"
escpi.cnam.fr   text = "Ecole Supérieure de Conception et Production
                                                            Industrielle "
escpi.cnam.fr   nameserver = pollux.escpi.cnam.fr
escpi.cnam.fr   nameserver = bellatrix.escpi.cnam.fr
escpi.cnam.fr   nameserver = asimov.cnam.fr
escpi.cnam.fr
      origin = pollux.escpi.cnam.fr
      mail addr = Postmaster.escpi.cnam.fr
      serial = 2000042020
      refresh = 21600 (6H)
      retry   = 3600 (1H)
      expire  = 1209600 (2W)
      minimum ttl = 86400 (1D)
escpi.cnam.fr   nameserver = pollux.escpi.cnam.fr
escpi.cnam.fr   nameserver = bellatrix.escpi.cnam.fr
escpi.cnam.fr   nameserver = asimov.cnam.fr
postman.escpi.cnam.fr   internet address = 163.173.112.5
fermi.cnam.fr   internet address = 163.173.128.60
pollux.escpi.cnam.fr    internet address = 163.173.112.100
bellatrix.escpi.cnam.fr internet address = 163.173.112.4
asimov.cnam.fr  internet address = 163.173.128.6
```

6) DNS: Approfondissements Administration du DNS

A) Utiliser le serveur DNS de son FAI

(**FAI** Fournisseur d'accès Internet)
(**ISP** 'Internet Service Provider').

S'il accepte de vous héberger correctement.
Si vous n'avez pas d'équipe système forte.

B) Construire son propre serveur

a) **Gérer son propre serveur** permet de **protéger** son architecture et éviter de donner des informations personnelles à gérer.

b) **On peut modifier, ajouter, détruire** les informations de son propre domaine sans en référer en permanence à son fournisseur.

c) La résolution des requêtes peut-être **plus rapide et plus efficace** que sur un serveur d'ISP surchargé.

Les trois étapes principales avec un fournisseur d'accès Internet

a) **Enregistrer un nom de domaine** auprès de l'autorité compétente (un nom .com .fr)

b) Le FAI vous donne les adresses IP de ses serveurs primaires et secondaires.

->Avec ces adresses on configure **toutes les piles TCP/IP** des sites clients DNS ou on **renseigne le serveur DHCP** 'Dynamic Host Configuration Protocol'.

c) Le FAI insère dans la base de données les enregistrements ('Resources Records') à rendre publics.

- **Pour recevoir du courrier** sur votre serveur de courrier (Mail Exchange MX).

- **Pour associer les adresses IP aux noms de domaines** (type A) serveur de courrier, serveur FTP, serveur WEB ainsi que toutes les machines à faire connaître.

La gestion de son propre serveur

L'administrateur d'un domaine :

- **Fixe les limites** des zones administrées.
- **Créé des fichiers** pour le serveur primaire.
- **Entretient** cette version locale.

a) L'administrateur définit **au moins un site serveur DNS primaire et au moins un site serveur secondaire** (obligatoires).

b) Il **créé et administre les informations nécessaires** sous forme de fichiers.

En Unix :

Fichier du site client DNS : /etc/resolv.conf

Fichier du site serveur DNS : /etc/named.boot

Administration du site client DNS (Unix)

Le fichier /etc/resolv.conf contient les lignes de directives :

- **domain** : le nom de domaine par défaut (cette chaîne de caractère complète les noms qui ne sont pas terminés par un point).
- **nameserver** : l'adresse IP d'un serveur DNS de la zone courante.
- **order** : l'ordre utilisé par le résolveur.
- **search** : la liste des domaines successifs à visiter si la stratégie de visite par défaut ne convient pas.

Exemple de fichier resolv.conf

```
$ cat /etc/resolv.conf
domain     cnam.fr
nameserver 163.173.128.6
nameserver 163.173.128.60
order local,bind
```

Administration du site serveur DNS (Unix)

Le fichier principal serveur **/etc/named.boot** définit les principaux fichiers utilisés pour :

. **Les bases de données** DNS des zones pour lesquels les serveurs sont primaires ou secondaires.

. **Les serveurs** au niveau **racine**.

Type des enregistrements de /etc/named.boot

- **directory** : définit le répertoire contenant les fichiers bases de données DNS à utiliser par le serveur.

directory nom_de_repertoire

Type des enregistrements de /etc/named.boot

primary : une zone pour laquelle ce serveur est un serveur primaire associé au nom d'un fichier de 'resource records' définissant la base de données DNS pour la zone.
Exemple : **primary** nom_de_zone fichier

secondary : le nom de la zone pour laquelle ce serveur est secondaire, au moins une adresse IP de serveur primaire, un nom du fichier utilisé par le serveur pour stocker les données DNS.
Exemple : **secondary** zone serveur fich

cache : définit le fichier qui contient les noms et les adresses IP des serveurs racines. Il sert à initialiser le cache du serveur (on peut le télécharger à partir du site) :
ftp://rs.internic.net/netinfo/root-servers.txt:

cache nom_de_fichier

DNS Conclusion (1)

Avantages

a) Un système **d'annuaire** distribué au niveau mondial qui remplit très bien son rôle initial

b) Diverses autres **possibilités** intéressantes (alias, pointeurs …)

c) Une organisation en **arbre** simple et efficace.

d) **Très utilisé** (standard de facto), **très rodé**.

e) Les problèmes de **performances** et de **sûreté** sont résolus par les serveurs **primaires** et **secondaires** mais aussi par l'utilisation de **caches**.

DNS Conclusion (2)

Inconvénients

- Faiblesse des mécanismes **de sécurité** dans la version de base: correction en cours de déploiement, nombreuses **RFC DNS sécurisé.**

- **D'autres besoins** comme la distribution de nouveaux types d'informations nécessaires aux utilisateurs ne sont pas couverts

Exemple : identificateurs d'usagers 'login', dictionnaires de données, …

=> Le DNS n'est pas très facilement adaptable.

=> Emergence de plusieurs autres systèmes **de serveurs d'annuaires** (LDAP, NIS+, UDDI, …).

DNS Conclusion (3)

Inconvénients

- Problèmes multiples dans la gestion des noms de domaines surtout ceux de plus haut niveau :

Les nouveaux suffixes adoptés après plusieurs années de débats:

.biz - affaires
.name - individuels
.museum - musées
.aero - aviation
.coop - cooperatif
.info – information

.pro - professionnel ? ?

DNS Biliographie

L. Toutain, ‘Réseaux locaux et Internet’, Hermes

W.R. Stevens ‘TCP/IP illustrated : the protocols’ Addison Wesley

A.S. Tannenbaum ‘Computer networks’ Prentice Hall.

II L'accès à distance

Exemple du protocole Telnet

Introduction
1 Architecture d’un accès distant
2 Telnet
- **Le terminal virtuel de Telnet**
- **La négociation**

Conclusion

Introduction : Notion de protocole de session à distance (‘Remote Login’)

- Un protocole de **session à distance** permet d'ouvrir un dialogue interactif sur un autre ordinateur au moyen d'un réseau.

- L’usager réalise un **échange bidirectionnel avec un processus distant** (une application ou un interpréteur de commandes) qui tourne sur le système distant.

- La communication est orientée **flot d’octets** pour interfacer des terminaux **caractères et des processus**. Autre possibilité : dialogue terminal/ terminal ou processus/ processus.

- Problème non traité dans ce chapitre : le support **distant** des terminaux multi **fenêtres**.

L’accès à distance : **L’ancêtre de toutes les applications réparties** (‘The mother of all application protocols’).

Les protocoles de session à distance dans l'Internet

Nombreuses implantations dans les architectures de réseau (en Arpanet/Internet depuis 1972). Actuellement deux protocoles.

TELNET

'TELecommunication NETwork' 1980

⇨ Solution très générale

- Utilisable pour tout type de système.
- Beaucoup de facilités de négociation d'options pour échanger les caractéristiques de chaque site communicant.

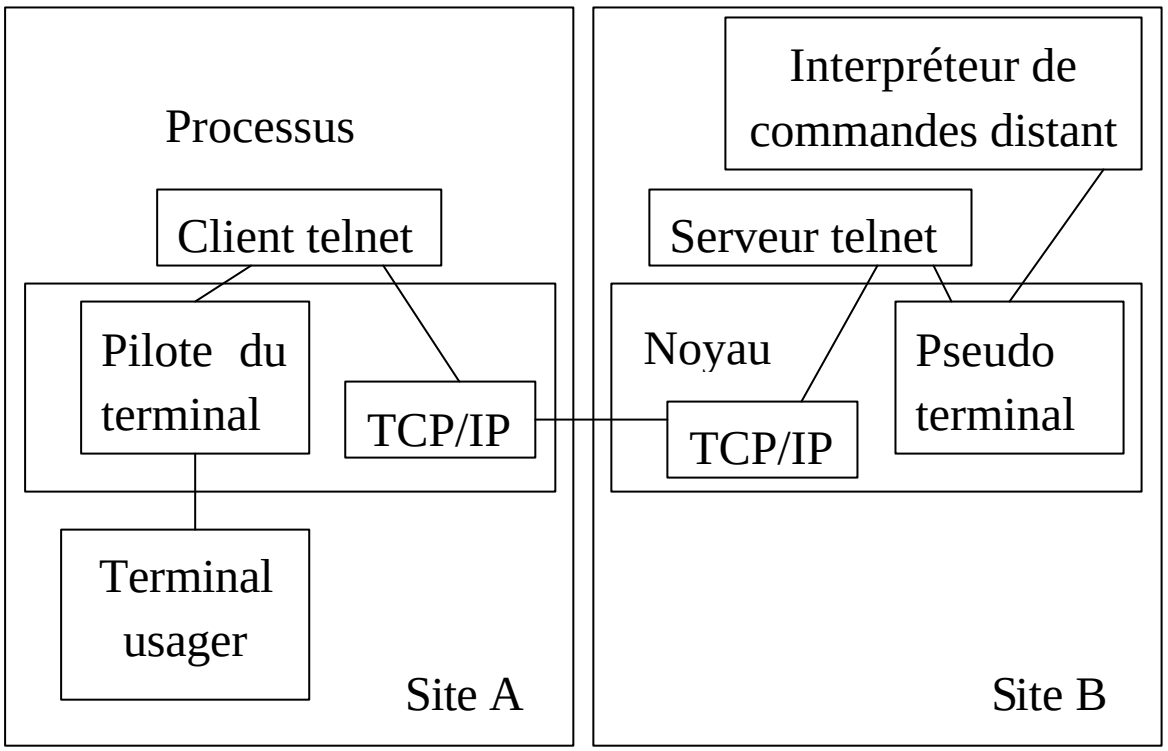
RLOGIN

'Remote LOGIN'

⇨ Solution spécifique des systèmes UNIX

- **Une seule option** retenue (celle de UNIX).
- Pas de négociation.
- Rlogin a néanmoins été porté sur différents autres systèmes qu'Unix.

1) Architecture générale d'un client serveur d'accès distant



Logiciels et matériels utilisés

- Terminal utilisateur local
- Pilote du terminal local
- Client d'accès distant ('telnet')
- Protocole réseau (TCP/IP)
- Serveur d'accès distant ('telnet')
- Pseudo terminal sur le site serveur distant.
- Processus exécuté à distance (par exemple interpréteur de commandes)

Explications du fonctionnement

- Le client telnet transmet les caractères entrés sur le **terminal** local **vers** le **serveur** distant.

- Le **serveur** distant ‘telnet’ transmet les **caractères** à un '**pseudo terminal**', analogue d'un pilote de terminal sur le site distant.

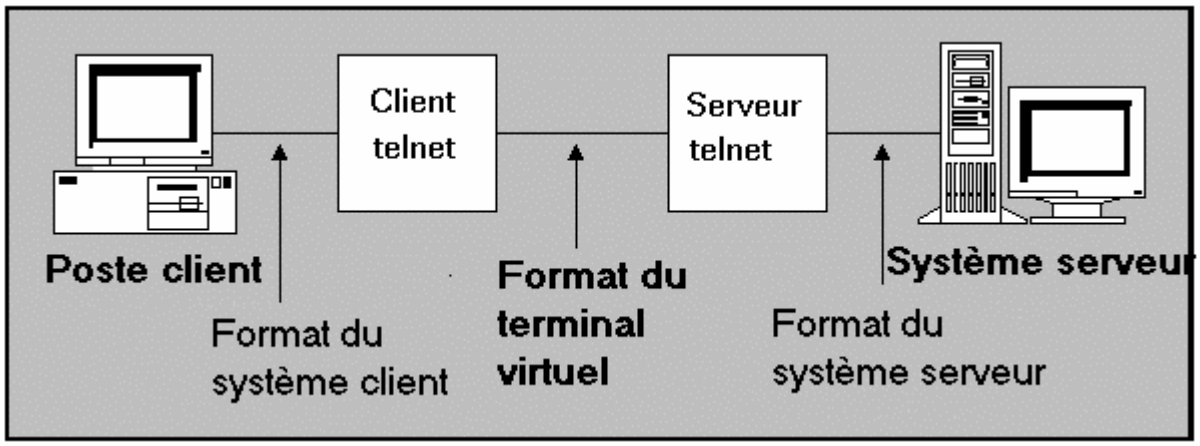
- Le fonctionnement est **bidirectionnel**: on supporte le même échange dans les deux sens.

- Sur la connexion réseau, il faut **émettre** les **caractères usager** (à transmettre à l’application) et les **directives de contrôle** de l'accès distant (les commandes telnet).

- Pour assurer l'**interopérabilité** on doit **convertir les formats d'un terminal** du client en format d'un terminal du site serveur.

- **L'interpréteur de commande est le même** que pour une connexion locale: il faut s'authentifier.

Notion de terminal virtuel réseau 'NVT Network Virtual Terminal'



- Les systèmes **hétérogènes** considèrent les périphériques de façon différente
=> **Notion d'appareil virtuel** (format pivot).

- Pour les terminaux : format de terminal réseau => **Notion de terminal virtuel**

.Choix d'un ensemble **d'options communes** pour gérer les terminaux (gestion de clavier, d'écran, … etc

. Un format nouveau **inexistant** ('**virtuel**').

. Quelquefois un format **existant** de préférence très répandu (ex : DEC VT100).

Fonctionnement du terminal virtuel réseau

- Définir comment fonctionne le protocole client serveur d'accès distant :

. **Les commandes du terminal virtuel** (les PDU du protocole)

Ex : Négocier des options

Tuer un processus

. **Les données échangées** .

Ex : Les caractères imprimables, les caractères de contrôles des terminaux

- **Clients et serveurs** d'accès distant traduisent commandes et données du format de leurs terminaux locaux vers le format terminal virtuel réseau (dans les deux sens).

Notion de négociation

- Selon les systèmes et les types de terminaux de **nombreuses variantes sont possibles** :
  . Beaucoup de fonctionnalités.
  . Fonctionnement **rustique** qui ne peut supporter des dialogues complexes.
  . **Certaines options** doivent être fixées : exemple qui fait l'écho.

- Une **négociation préliminaire** permet à un système de sélectionner le profil des échanges en ne se limitant pas au mode le plus simple.

- **Négociation symétrique** : de préférence les deux entités communicantes (client et serveur) doivent avoir les mêmes droits dans la négociation. On peut avoir des modes processus/processus ou terminal/processus.

2) Exemple de TELNET

Le terminal virtuel de Telnet
Le NVT 'Network virtual terminal'

A) Le jeu de caractères du NVT

L'ensemble des **caractères imprimables** et ensemble de **caractères de contrôles**.

- Pour les **caractères imprimables** jeu de caractère pivot: **l'US-ASCII** (95 caractères imprimables).

- Pour les **caractères de contrôle** définition des opérations conceptuelles caractéristiques des terminaux caractères.

Ex: **notation** d'un caractère de contrôle : cr
signification : retour chariot.

- Associer aux différents caractères de contrôle **un code caractère échangé**.

Ex: cr valeur décimale 13.
(33 codes également en format 7 bits).

Table des principaux caractères de contrôle du NVT Telnet

ASCII Code	Valeur Décimale	Signification
NUL	0	Caractère nul (No operation : has no effect on output).
BEL	7	Cloche (Sound audible/visible signal : no motion).
BS	8	Retour arrière (Backspace : move left one character position).
HT	9	Tabulation horizontale (Move right to the next horizontal tab).
LF	10	Saut de ligne (Move down vertically to the next line).
VT	11	Tabulation verticale (Move down to the next vertical tab stop).
FF	12	Saut de page (Move to the top of the next page).
CR	13	Retour chariot (Move to the left margin on the current line).

B) Les commandes Telnet

Echangés sous forme de suites d'octets dans le même flot que les données usagers (**signalisation dans la bande**)

a) **Commandes de base** : deux octets

< IAC > < COMMAND >

IAC = 'Interpret As Command' (code = 255)

b) **Négociations** : séquence de trois octets

< IAC > < COMMAND > <OPTION >

c) **Mode transparent** : pour l'octet code 255

< IAC > < IAC >

Remarque: NVT utilise le standard 7-bit US-ASCII pour les données (bits de poids fort à 0). Il réserve les octets avec bit de fort poids à 1 pour coder les commandes Telnet.

Liste des commandes Telnet

Nom	code décimal	Description
EOF	236	Fin de fichier (end-of-file)
SUSP	237	Suspension du processus en cours
ABORT	238	Destruction du processus en cours
EOR	239	Fin de l'enregistrement
SE	240	Fin de la sous-option
NOP	241	Opération vide
DM	242	Marque de données
BRK	243	Break
IP	244	Interruption de processus
AO	245	Fin de sortie immédiate
AYT	246	Etes-vous là?
EC	247	Caractère d'échappement
EL	248	Effacement de ligne
GA	249	Retournement (demi duplex)
SB	250	Début de sous-option
WILL	251	Négociation d'option
WONT	252	Négociation d'option
DO	253	Négociation d'option
DONT	254	Négociation d'option
IAC	255	Octet de début de commande

Quelques compléments sur les commandes

Abort : tuer le processus en cours.

Abort Output (AO): Supprimer les sorties en attente.

Are You There (AYT): Vérifier que le système à distance est opérationnel.

Erase Character (EC): Destruction du dernier caractère émis.

Erase Line (EL) : Destruction de la dernière ligne.

Go Ahead (GA) : Retournement de sens (‘Turn over control’), pour terminaux half-duplex.

Data Mark (DM) : Marque dans le flot de données. En cas de problème, émission d’un signal SYNC en mode urgent TCP : toutes les données en cours sont ignorées jusqu’au DM.

Principe de la négociation en Telnet

- Sélectionner **des options** (choix relatifs au terminal) **différentes** des valeurs par défaut.

Exemple: écho local ou à distance.

- La négociation est **symétrique** (elle peut être initialisée par les deux cotés).

Les quatre commandes de négociation

WILL <option> l'émetteur veut utiliser une option (localement).

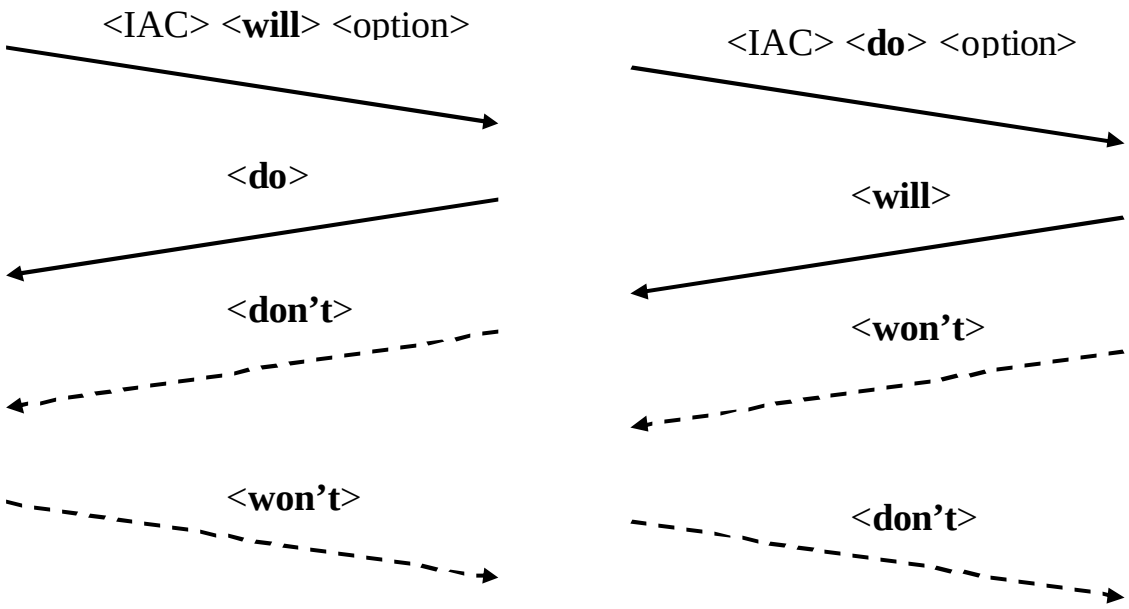
DO <option> l'émetteur requiert l'utilisation de l'option dans le site distant.

WON'T <option> rejet d'utilisation d'une option par le site distant.

DON'T <option> refus d'une option proposée par le site distant pour le site local.

Remarque : Pour certaines options assez spécialisées (plusieurs valeurs, volume d'information) il existe une notion de sous option ('suboption').

Fonctionnement de la négociation Telnet



Rappel

Will - veut utiliser.
Do - veut que l'autre utilise.
Won't - refuse d'utiliser.
Don't - refuse que l'autre site utilise.

Quelques Echanges

Will -> **Do** Propose pour lui et l'autre accepte
Do->**Will** Propose pour l'autre,l'autre accepte
Will -> **Don't** Refus d'utiliser par l'autre
Do -> **Won't** Refus d'utiliser par lui.
Don't -> **Won't** L'autre veut supprimer.
Won't->**Don't** Le local veut supprimer.

Exemples de négociation

a) Un coté veut utiliser un **mode binaire.** Il émet une commande **WILL Binary.** Le distant répond soit **DO Binary** Transmission (acquittement positif) soit **DON'T Binary** Transmission (rejet).

b) Un terminal ne veut pas faire l'écho: il émet **WON'T Echo**. Le site distant est d'accord il répond **DON'T Echo**.

c) Autre négociation possible: le **contrôle de fl**ux des terminaux (**XON/XO**FF) peut être traité localement (par le client Telnet) ou par le serveur (RFC 1080).

d) Autre négociation possible: utilisation d'un code autre que l'USASCII (RFC 1647).

Début de liste des options négociables

Liste du Code Nature de l'Option Références de la RFC

0 Binary Transmission RFC 856

1 Echo RFC 857

2 Reconnection

3 Suppress Go Ahead RFC 858

4 Approx Message Size Negotiation

5 Status RFC 859

6 Timing Mark RFC 860

7 Remote Controlled Trans and Echo RFC 726

8 Output Line Width

9 Output Page Size

10 Output Carriage-Return Disposition RFC 652

11 Output Horizontal Tab Stops RFC 653

12 Output Horizontal Tab Disposition RFC 654

13 Output Formfeed Disposition RFC 655

14 Output Vertical Tabstops RFC 656

15 Output Vertical Tab Disposition RFC 657

16 Output Linefeed Disposition RFC 657

17 Extended ASCII RFC 698

18 Logout. RFC 727

19 Byte Macro RFC 735

20 Data Entry Terminal RFC 732, RFC 1043

21 SUPDUP RFC 734, RFC 736

22 SUPDUP Output RFC 749

23 Send Location RFC 779

24 Terminal Type RFC 1091

Début de la liste des RFC de négociation (2)

25 End of Record RFC 885

26 TACACS User Identification RFC 927

27 Output Marking RFC 933

28 Terminal Location Number RFC 946

29 Telnet 3270 Regime RFC 1041

30 X.3 PAD RFC 1053

31 Negotiate About Window Size RFC 1073

32 Terminal Speed RFC 1079

33 Remote Flow Control RFC 1372

34 Linemode RFC 1184

35 X Display Location RFC 1096

36 Environment Option RFC 1408

37 Authentication Option RFC 1416, RFC 2941, RFC 2942, RFC 2943, RFC 2951

38 Encryption Option RFC 2946

39 New Environment Option RFC 1572

40 TN3270E RFC 2355

41 XAUTH

42 CHARSET RFC 2066

43 Telnet Remote Serial Port (RSP)

44 Com Port Control Option RFC 2217

45 Telnet Suppress Local Echo

46 Telnet Start TLS

47 KERMIT RFC 2840

. . . .

Exemple de trace du fonctionnement du protocole de négociation Telnet

```
1    % telnet
2    telnet> toggle options
3    Will show option processing.
4    telnet> open pauli
5    Trying 163.173.129.19...
6    Connected to pauli.cnam.fr.
7    Escape character is '^]'.
8
9    SENT do SUPPRESS GO AHEAD
10   SENT will TERMINAL TYPE
11   SENT will NAWS
12   SENT will TSPEED
13   SENT will LFLOW
14   SENT will LINEMODE
15   SENT do STATUS
16   SENT will 36
17   RCVD do TERMINAL TYPE
18   RCVD do TSPEED
19   RCVD do 35
20   SENT wont 35
21   RCVD do 36
22   RCVD will SUPPRESS GO AHEAD
23   RCVD do NAWS
24   Sent suboption NAWS 0 80 (80) 0 24 (24)
25   RCVD do LFLOW
26   RCVD dont LINEMODE
27   RCVD will STATUS
28   RCVD IAC SB
29   Received suboption TERMINAL-SPEED SEND
30   Sent suboption TERMINAL-SPEED IS 9600,9600
31   RCVD IAC SB
32   Received suboption ENVIRON SEND
33   Sent suboption ENVIRON IS VAR "USER" VALUE"gerard"
34   RCVD IAC SB
35   Received suboption TERMINAL-TYPE SEND
36   Sent suboption TERMINAL-TYPE IS "VT220"
37   RCVD do ECHO
38   SENT wont ECHO
39   RCVD will ECHO
40   SENT do ECHO
41
42   Digital UNIX (pauli) (ttyq0)
43
```

44 login:

Commentaires de la trace

- Lignes 1 à 7 : Lancement d'un client Telnet sur la machine Pauli avec sélection toggle option de trace de la négociation des options.
- Ligne 9 SENT do SUPPRESS GO AHEAD : Demande du client au serveur de ne pas utiliser GA commande de retournement de sens en half duplex.
- Ligne 10 SENT will TERMINAL TYPE : Demande d'échange du type de terminal. Demande acceptée ligne 17 et réalisée ligne 35 et 36.
- Ligne 11 SENT will NAWS : Veut dire Negociate About Window Size. La commande est acceptée ligne 23 et ligne 24 les paramètres sont envoyés sous forme d'une sous option. Il s'agit du nombre de lignes et de colonnes affichées.
- Ligne 12 SENT will TSPEED : Demande l'échange des vitesses des terminaux. Accepté ligne 18 et réalisé lignes 29 et 30.
- Ligne 13 SENT will LFLOW : Indique que le client veut utiliser un contrôle de flux local contrôle Q contrôle S(Local Flow Control) . Le serveur accepte ligne 25.
- Ligne 14 SENT will LINEMODE : Demande au serveur d'émettre des lignes complètes. La commande est rejetée ligne 26. Les échanges se feront caractères par carctères.
- Ligne 15 SENT do STATUS : Demande de pouvoir échanger avec le serveur le status c'est à dire l'état des options négociées. Accepté ligne 27.
- Ligne 16 SENT will 36 : Les options peuvent être codées sous la forme d'entiers en décimal. L'option 36 est l'option associées à l'échange de variables d'environnements. Elle est acceptée ligne 21 et son effet apparaît lignes 31 à 33 ou est échangée comme variable d'environnement le nom de l'usager.
- Ligne 19 RCVD do 35 : Demande de négociation pour affichage X11 . Rejettée ligne suivante 20.
- Lignes 28 RCVD IAC SB : Indication de début de négociation de sous option (même chose en 31 et 24).
- Lignes 37 RCVD do ECHO ; 38 SENT wont ECHO ; 39 RCVD will ECHO ; 40SENT do ECHO Négociation de l'extrémité qui effectue l'écho des caractères. Le site local (le client) refuse. L'écho sera effectué à distance c'est à dire par le serveur.

- Lignes 41 à 44 : On arrive au login sur la machine distante.

Telnet sur TCP/IP

Le client Telnet

- En UNIX, le client “telnet” est lancé par la commande :
  telnet <site distant> <numéro de port>

- Création d’une connexion TCP avec le serveur “telnet” de la machine distante.

- Utilisation du **port 23** pour l’accès à un interpréteur de commande.

- Utilisation d’un **autre port** pour une autre application qui utiliserait telnet.

- Le client **accepte les lignes** de commande ou les **données à envoyer** à distance et **les données en provenance** du serveur.

Exemple de la liste des commandes disponibles (sous Unix)

```
telnet> help
Commands may be abbreviated.  Commands are:
close       close current connection
logout      forcibly logout remote user and close the
connection
display     display operating parameters
mode        try to enter line or character mode
                                          ('mode ?'for more)
open              connect to a site
quit        exit telnet
send        transmit special characters ('send ?' for more)
set         set operating parameters ('set ?' for more)
unset       unset operating parameters ('unset ?' for more)
status      print status information
toggle      toggle operating parameters ('toggle ?' for more)
slc         change state of special charaters ('slc ?' for more)
z           suspend telnet
!           invoke a subshell
environ     change environment variables ('environ ?' for more)
?           print help information
<return>    leave command mode
telnet>
```

Le serveur Telnet

- Le serveur s'exécute sur la **machine distante** : si le serveur n'est pas en train de **s'exécuter**, le service **n'est pas disponible**.

- En Unix on appelle **Démons** les serveurs qui s'exécutent sous la forme de processus en arrière plan.

- Un serveur prêt **attend une requête**.

- **Rend le service** demandé.

- **Renvoie le résultat** au client.

- **Se met à nouveau** en attente.

Conclusion : Accès distant Telnet

- Le protocole de sa catégorie **le plus utilisé**.

- Des implantations **portées dans tous les environnements.**

- A ce jour pratiquement une **centaine** de RFC concernent Telnet.

- Un protocole **ancien mais constamment remis à jour**.
  Exemple : intégration des protocoles d'authentification récents plus sécurisés.

- Un protocole avec différents domaines d'application outre **l'accès à distance :**
  . **Test** d'applications réparties
  . Fonctionnement en **manuel** des principaux protocoles d'application en mode caractère Internet (SMTP, HTTP).

- Le protocole le plus **sécurisé** d'accès à distance est ssh (secure shell).

Bibliographie

Internetworking With TCP/IP, Volume I / Douglas E. Comer Prentice-Hall

TCP/IP Illustrated W. Richard Stevens Addison Wesley

RFC 818: The Remote User Telnet Service

RFC 854: Telnet Protocol Specification

RFC 855: Telnet option specifications