

Internet et Outils
L1/IO2 2007 - 2008
S2-IO2
Introduction aux Réseaux

François Armand

armand@informatique.univ-paris-diderot.fr

Plan Général Prévisionnel (1/2)

(non contractuel)

• Cours Internet et Outils:

- [1/12] Intro, Internet, Web, XHTML (2H)
- [2/12] XHTML(2H)
- [3/12] CSS (2H)
- [4/12] PHP (2H)
- [5/12] PHP suite (2H)
- [6/12] Introduction MySQL, Table simple (2H)
- [7/12] ► Partiel (2H) 15 Mars 08 14H-16H ◀
Amphis 2A et 8C

Plan Général Prévisionnel (2/2)

(non contractuel)

- Bases de données
 - [8/12] Tables multiples, Schémas (2H)
 - [9/12] Concepts un peu plus avancés (2H)
 - [10/12] **Introduction aux réseaux(2H)**
 - [11/12] Réseaux: compléments (2H)
 - [12/12] Révisions (2H)

Unix, Réseau: Plan

- Ce que vous allez découvrir
- Usage des réseaux
- Problèmes à résoudre
- Protocoles réseaux
- Noms et Ports
- Services de nommage
- Transferts de fichiers
- Accès à distance
- Ce qu'il faut retenir

Ce que vous allez découvrir

- Manières de connecter des machines
- Désigner un destinataire:
 - Noms de machines, numéros de ports
- Parler un langage commun:
 - Protocoles
- Transférer des fichiers
 - ftp, scp,...
- Ouvrir une session sur une machine distante

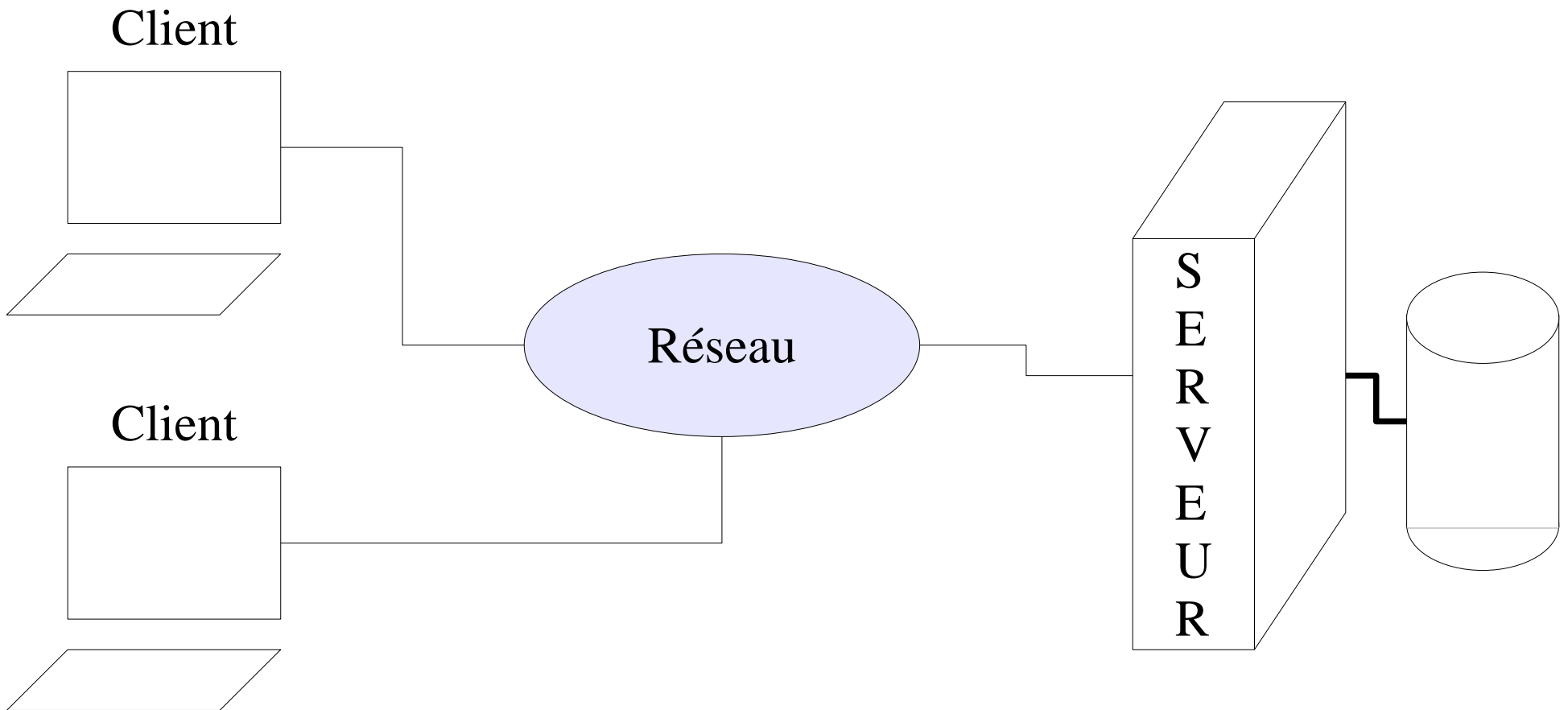
Unix, Réseau: Plan

- Ce que vous allez découvrir
- **Usage des réseaux**
- Problèmes à résoudre
- Protocoles réseaux
- Noms et Ports
- Services de nommage
- Transferts de fichiers
- Accès à distance
- Ce qu'il faut retenir

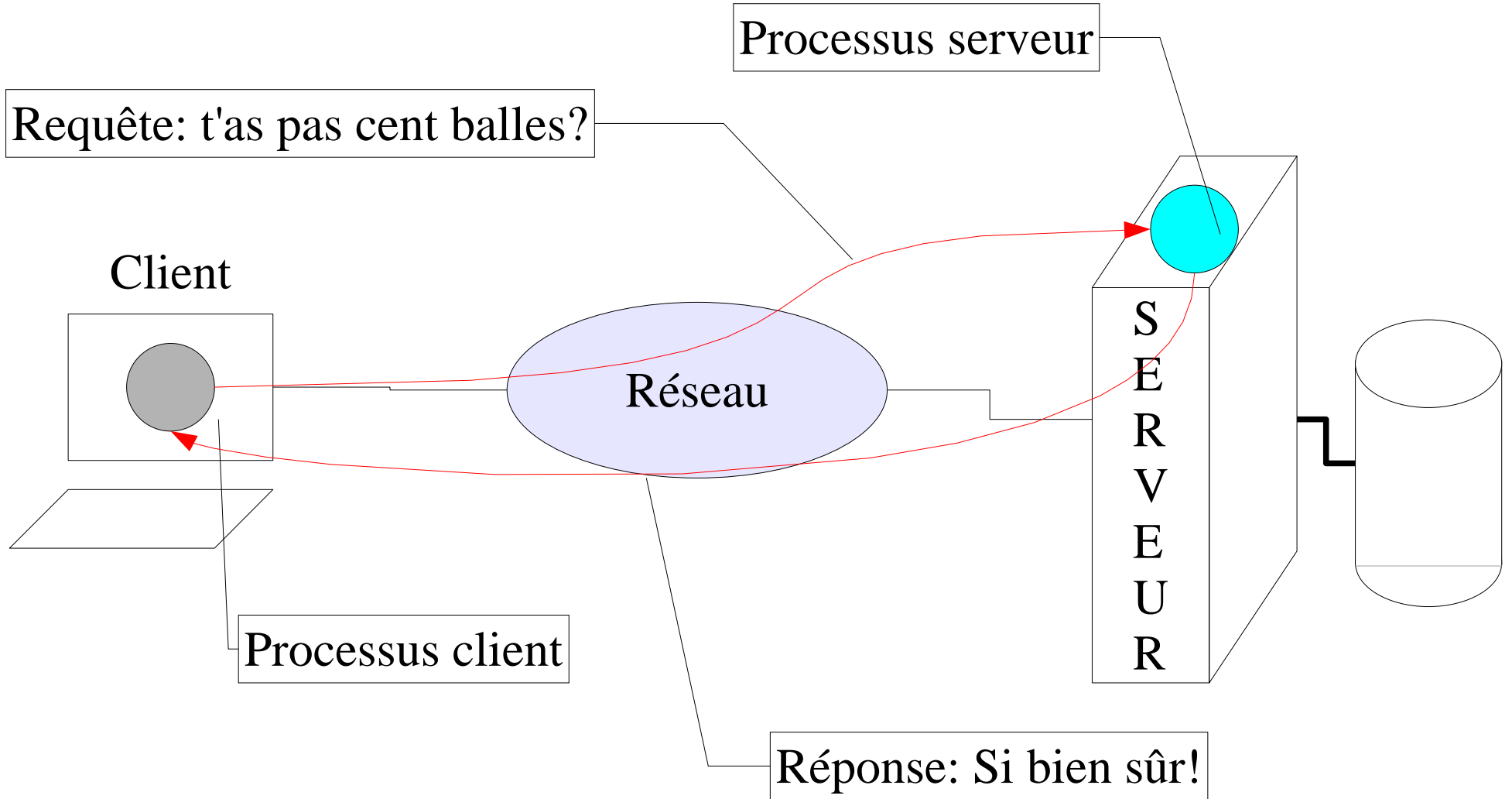
Usage des Réseaux

- Applications professionnelles
 - Partager des ressources
 - Ex: imprimantes, accès distant à une base de données,
 - Communication
 - Ex: mail, vidéo-conférence, commerce électronique
 - Modèle **Client-Serveur**
 - Clusters, grilles
 - Pair-à-pair (Peer-to-peer)

☰ Modèle Client - Serveur



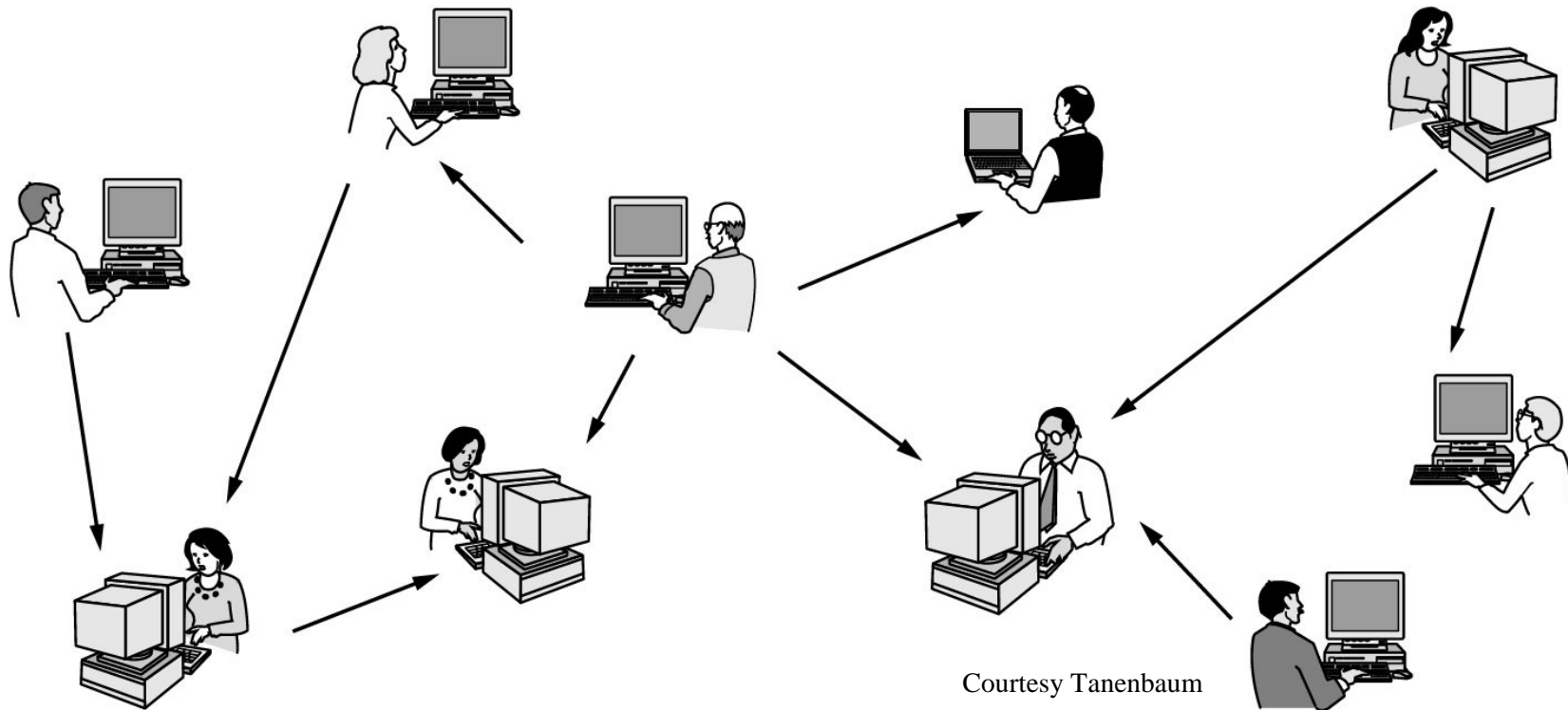
Modèle Client - Serveur



Usage des Réseaux

- Applications personnelles
 - Accéder à des informations distantes
Ex: Sites web
 - Communication
Ex: Courrier, Messagerie instantanée, Bavardage
 - Échanges
Ex: Peer-to-peer (poste à poste)
 - Jeux
 - Commerce électronique
 - Téléphonie, Télévision

Pair – à - Pair (*peer to peer*)



Unix, Réseau: Plan

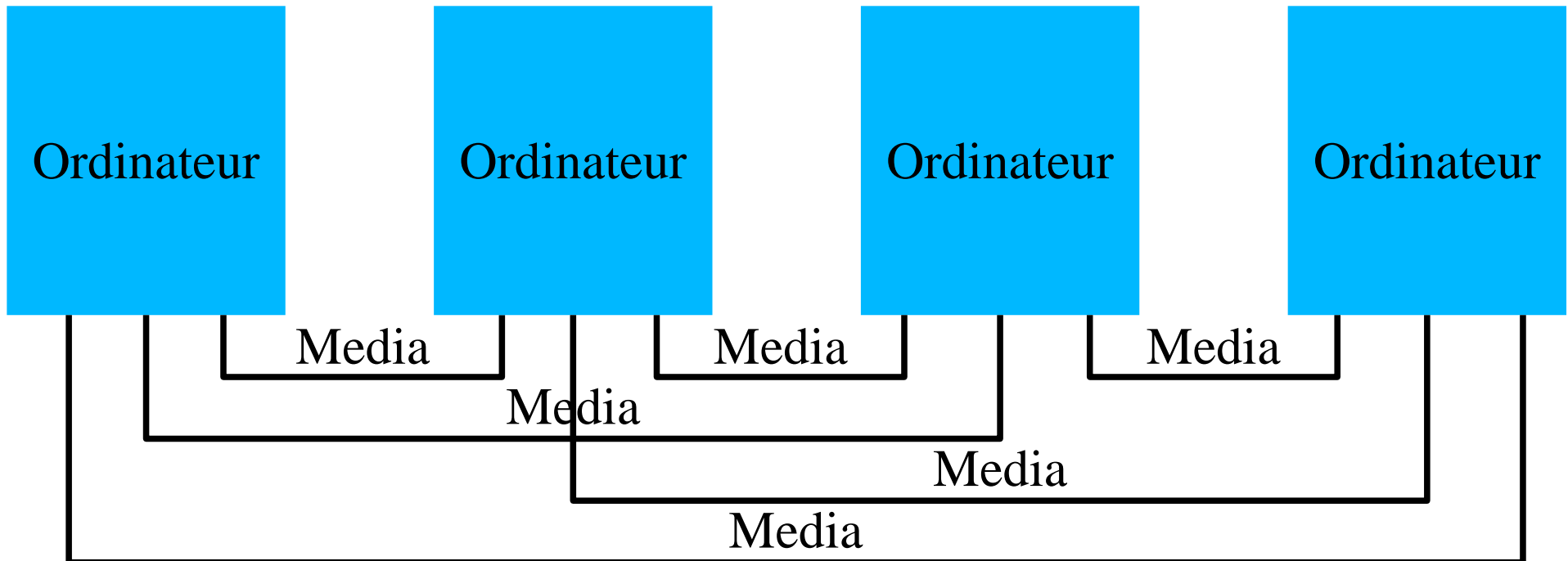
- Ce que vous allez découvrir
- Usage des réseaux
- **Problèmes à résoudre**
- Protocoles réseaux
- Noms et Ports
- Services de nommage
- Transferts de fichiers
- Accès à distance
- Ce qu'il faut retenir

Problèmes à résoudre

- **Connecter les machines**
 - Support physique,
 - Topologie (schéma de câblage)
- Tour de Babel:
 - Machines hétérogènes
- Partager les connexions entre plusieurs utilisations simultanées

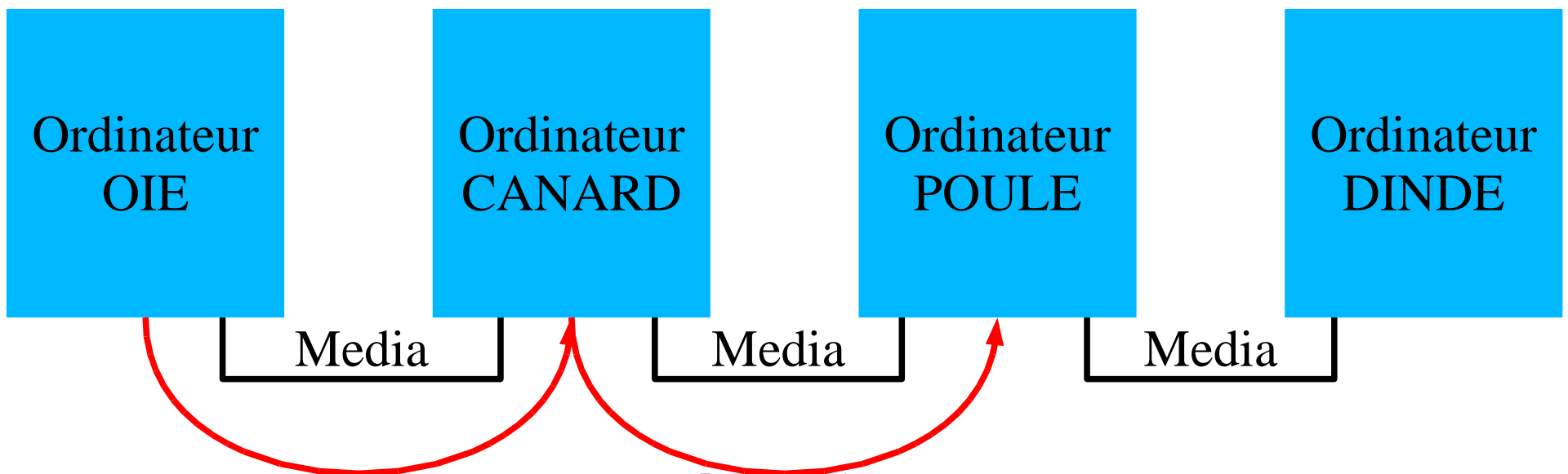
Liaison et Topologie

- Communication point à point entre plusieurs machines
 - Nombreuses liaisons, difficulté de câblage
 - Impossible si le nombre de machines est grand



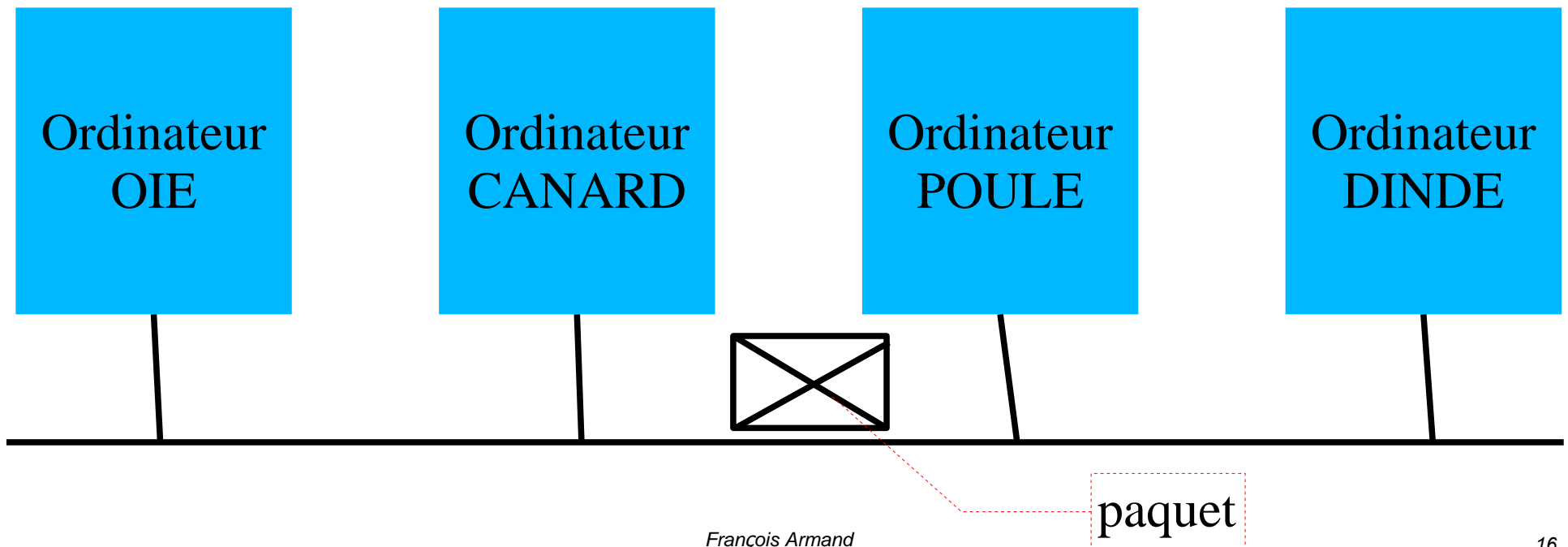
Liaison et Topologie

- Ne pas inter-connecter toutes les machines directement:
 - Pour envoyer une information de OIE à POULE, il faut la faire transiter par CANARD
 - Trouver le chemin entre émetteur et destinataire:
routage

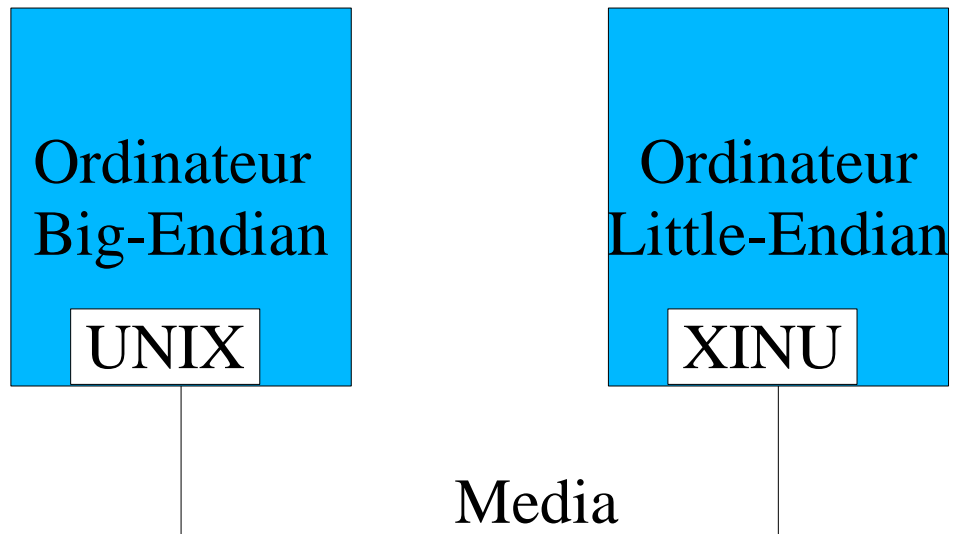


Liaison et Topologie

- Communication entre plusieurs machines
 - Réseau à diffusion
 - Câblage simple
 - Envoi direct d'une machine à toutes les autres



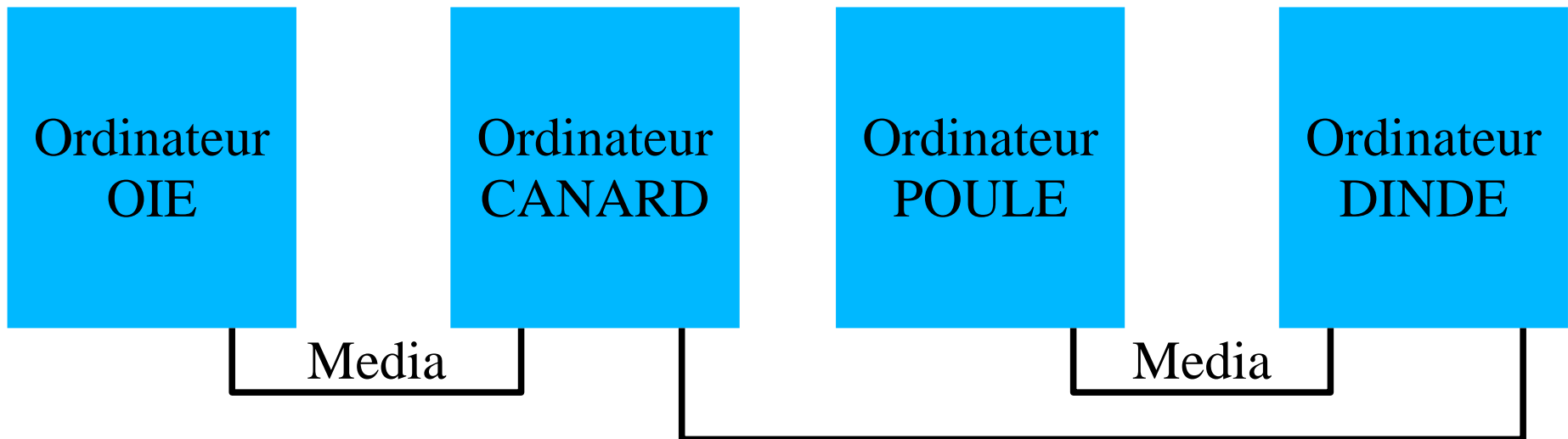
Échanges entre machines



- Données:
 - La chaîne UNIX sera vue comme XINU sur l'autre machine.
- Programmes:
 - Les jeux d'instructions différents, impossible de transmettre un programme exécutable (sauf interprété: Java)

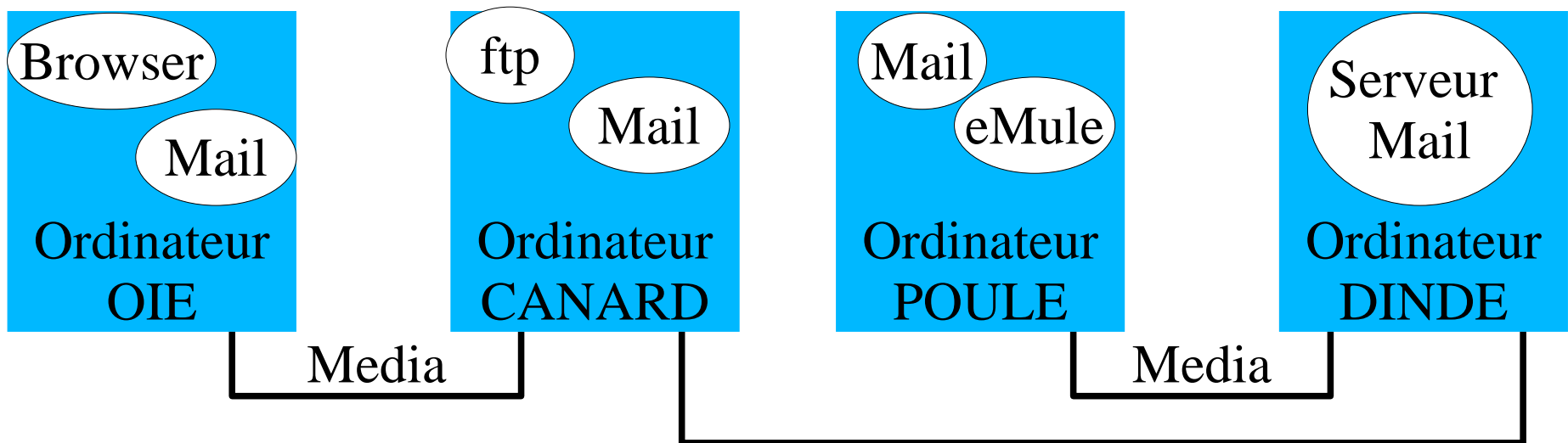
Concurrence entre machines

- DINDE peut recevoir et avoir à traiter des requêtes en provenance simultanée de CANARD (OIE) et POULE.



Concurrence entre applications

- Plusieurs applications accèdent au réseau simultanément (Ex: Mail, EMule sur POULE)
- Le serveur de Mail doit fournir son service à plusieurs clients simultanément



Unix, Réseau: Plan

- Ce que vous allez découvrir
- Usage des réseaux
- Problèmes à résoudre
- **Protocoles réseaux**
- Noms et Ports
- Services de nommage
- Transferts de fichiers
- Accès à distance
- Ce qu'il faut retenir

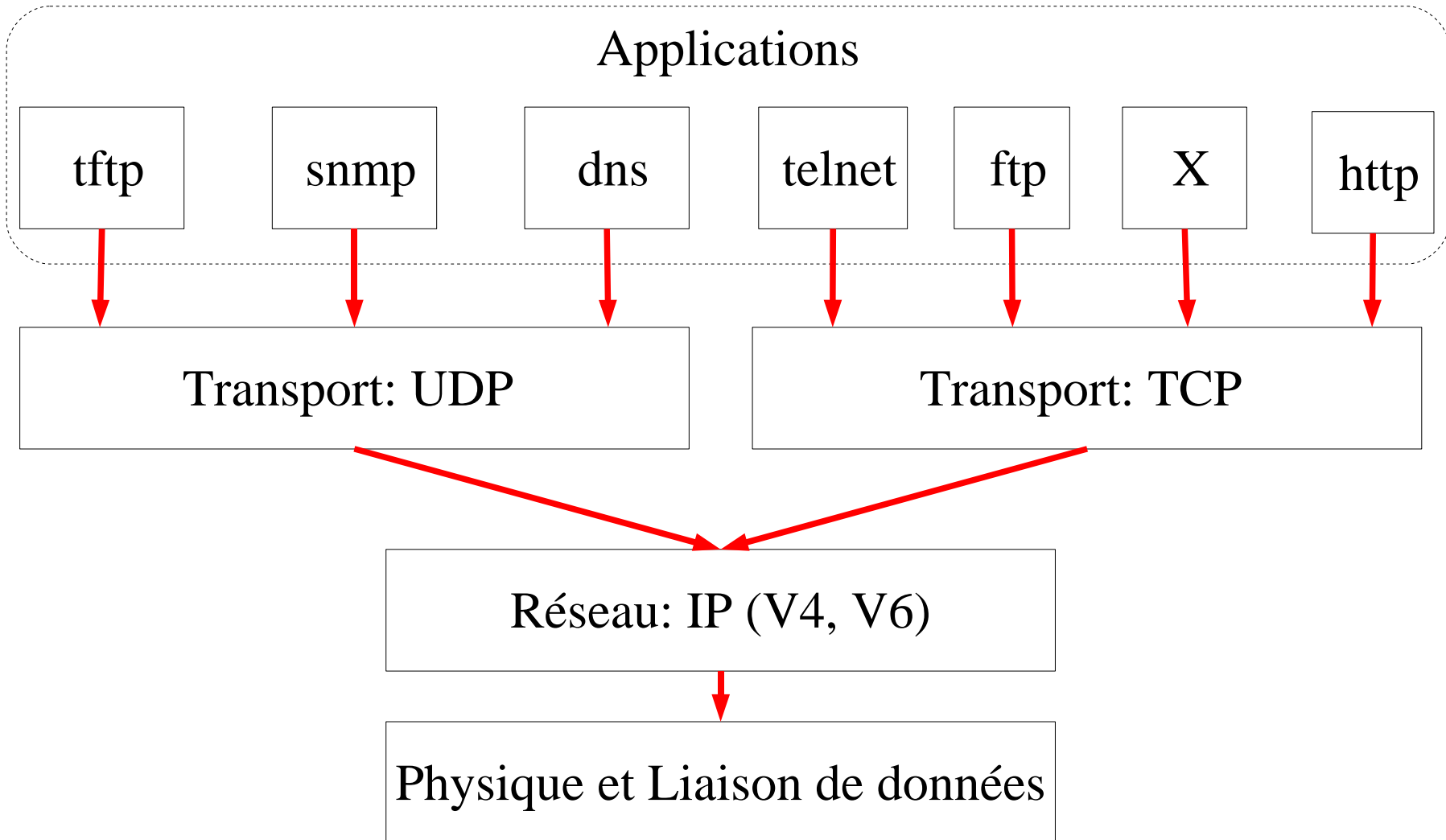
Logiciels Réseaux: Rôles

- Adressage: comment trouver le destinataire
- Contrôle d'erreur: Gérer les erreurs dues aux transmissions, gérer les pertes
- Contrôle de flux: Réguler le trafic
- Multiplexage: Gérer plusieurs demandes / transferts
- Routage: Acheminer les données à destination

Exemples de Protocole

- Humain:
 - Question: "How do you do?"
 - Réponse: "How do you do?"
- Entre machines:
 - Requête: "Envoyer la page web index.html"
 - Réponse: "<html> <head>..... </html>"
 - OU
 - Réponse: "Error 404"

Modèle TCP/IP



Unix, Réseau: Plan

- Ce que vous allez découvrir
- Usage des réseaux
- Problèmes à résoudre
- Protocoles réseaux
- **Noms et Ports**
- Services de nommage
- Transferts de fichiers
- Accès à distance
- Ce qu'il faut retenir

Adresses et Numéros de Port

- Chaque machine a en principe:
 - une adresse unique (dite adresse IP)
 - Ex (IPv4) : 192.168.1.1 , 127.0.0.1
 - ou plusieurs
 - les adresses sont affectées aux « interfaces », points d'attache de la machine au(x) réseau(x)
- Sur une même machine, chaque application attendant des requêtes du réseau est identifiée par un numéro, dit numéro de port.
 - ftp : 21; ssh : 22; telnet : 23; http : 80
 - tftp : 69; time: 37

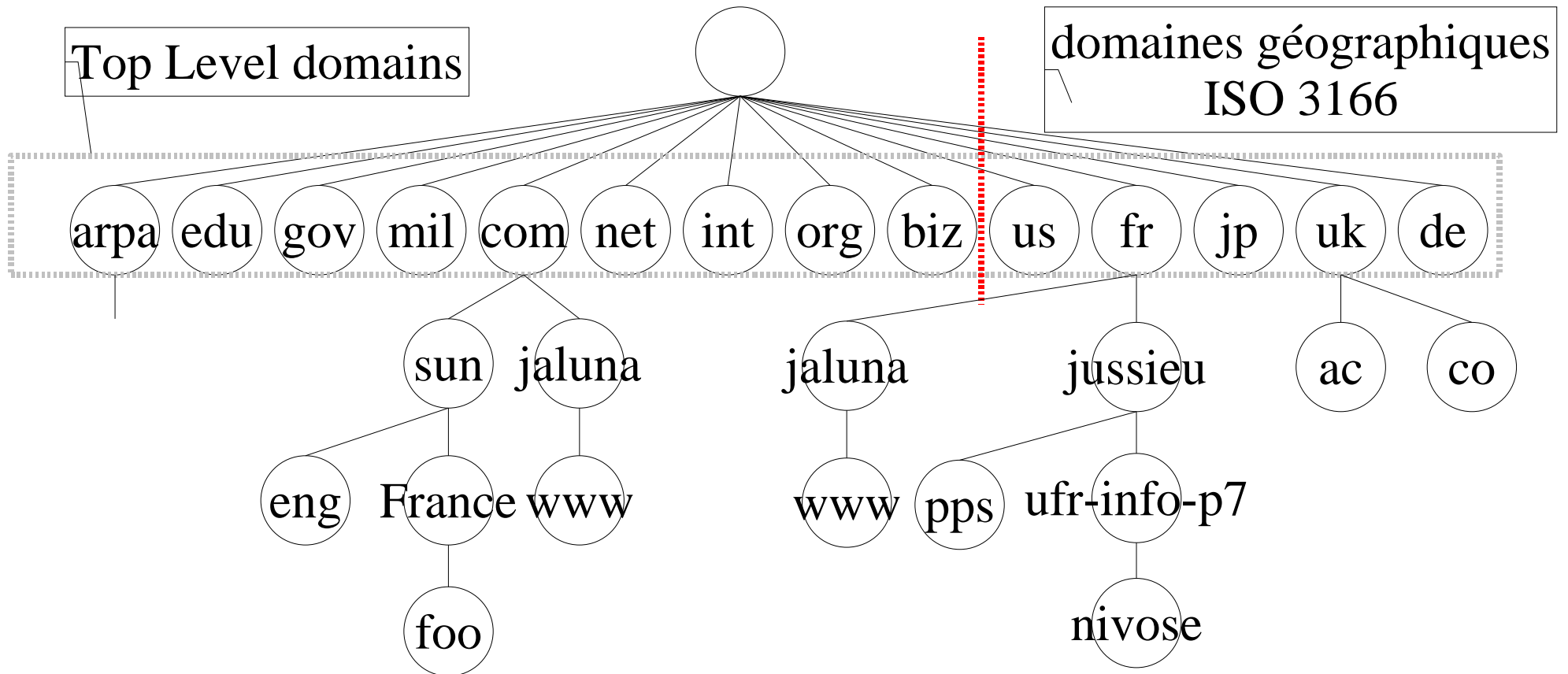
Noms symboliques

- Nommer les machines par un nom plutôt que par son adresse IP
 - Chaîne de caractères
 - Plus "naturel"
- Espace de noms hiérarchique plutôt que "plat"
 - Limiter les conflits
 - Décentraliser sa gestion, délégation d'autorité
- Résolution de nom en adresse IP

Unix, Réseau: Plan

- Ce que vous allez découvrir
- Usage des réseaux
- Problèmes à résoudre
- Protocoles réseaux
- Noms et Ports
- **Services de nommage**
- Transferts de fichiers
- Accès à distance
- Ce qu'il faut retenir

Espace de noms DNS



- Pas de correspondance avec la topologie physique du réseau
- DNS: *Domain Name System*

Résolution des Noms

- La conversion des noms en adresse est prise en charge par des serveurs de nommage
- Chaque serveur connaît une petite partie de l'ensemble des noms
- Les serveurs coopèrent entre eux pour convertir un nom en adresse

Consultation du Service de Nommage

- On peut interroger les serveurs de nommage

- `nslookup`

- Exemple:

```
--> nslookup
```

```
Default Server:  nivose.ufr-info-  
p7.jussieu.fr
```

```
Address: 132.227.70.73
```

```
>
```

Consultation du Service de Nommage

```
> www.freebsd.org
```

```
Server: nivose.ufr-info-p7.jussieu.fr
```

```
Address: 132.227.70.73
```

```
Non-authoritative answer:
```

```
Name: www.freebsd.org
```

```
Address: 216.136.204.117
```

```
>
```

Consultation du Service de Nommage

```
> jalap.xx.inex
```

```
Server:  nivose.ufr-info-  
p7.jussieu.fr
```

```
Address:  132.227.70.73
```

```
*** nivose.ufr-info-p7.jussieu.fr  
can't find jalap.xx.inex: Non-  
existent host/domain
```

```
>
```

Unix, Réseau: Plan

- Ce que vous allez découvrir
- Usage des réseaux
- Problèmes à résoudre
- Protocoles réseaux
- Noms et Ports
- Services de nommage
- **Transferts de fichiers**
- Accès à distance
- Ce qu'il faut retenir

Transferts de Fichiers

- FTP : *File Transfer Protocol*
- Suppose l'existence d'un démon sur la cible (`ftpd` ou `in.ftpd`)
- Invocation:
 - --> `ftp [-option] [hôte]`
 - option: `-i` : interactif, `-v` mode bavard, `-g`, `-n`
- S'appuie sur le protocole TCP/IP
- `sftp`: idem mais protocole de transfert sécurisé (chiffrement de `ssh`)

Transferts de Fichiers

```
neon--> ftp fluor
```

Nom global trouvé

```
Connected to fluor.pps.jussieu.fr.
```

```
220 fluor FTP server ready.
```

```
Name (fluor:armand): armand
```

```
331 Password required for armand.
```

```
Password:****
```

```
220 User armand logged in.
```

```
Remote system type is UNIX.
```

```
Using binary mode to transfer files.
```

```
ftp>
```

FTP: Gestion de la Connexion

- `ascii / binary` : type du transfert
- `bye / quit` : pour terminer la commande ftp
- `close / disconnect`: termine la connexion, pas ftp
- `glob`: expansion des noms de fichiers
- `open hôte [port]`: connexion vers l'hôte
- `prompt`: mode interactif / non-interactif
- `user nom`: identification sur machine distante

FTP : commandes

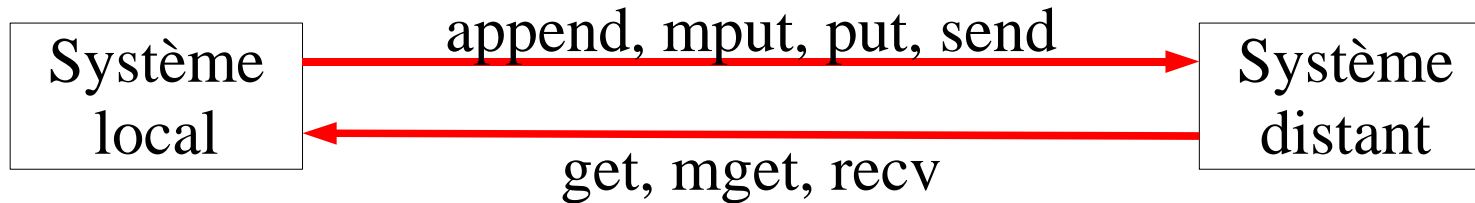
- Locales:

- `![cmd]` : exécution de `cmd` ou du shell sur machine locale
- `lcd [répertoire]`: répertoire courant local

- Distantes:

- `cd [répertoire]`: répertoire courant distant
- `delete référence`: suppression sur machine distante
- `ls [répertoire]`: contenu du catalogue distant
- `mdelete référence...` : delete multiple
- `rmdir répertoire`: suppression sur machine distante

FTP : Transferts



- append [local [distant]]
- put | send [local [distant]]
- mput local...
- get | recv [distant [local]]
- mget distant...



Autres commandes de transfert

- scp: remplace rcp

scp [-option] ... [[nom1]@hote1:]fichier [[nom2]@hote2:]ref

- permet de copier depuis une machine 3 un fichier ou un répertoire (de manière récursive) d'une machine 1 vers une machine 2
- rsync:
 - permet de synchroniser deux arborescences entre elles en réduisant le volume des données transférées aux différences entre les 2 arborescences

Transfert par HTTP: URL

- Forme générale: URL
 - Uniform Resource Locator (IETF RFC 1738 -1994)
 - Parfois appelé "Universal" au lieu de Uniform
- **<schéma>:<partie spécifique au schéma>**
- schémas usuels:
 - http, file, ftp, mailto, telnet..
- URI Uniform Resource Identifier
 - IETF STD 0066 ou RFC 3986

Uniform Resource Locator

- Partie spécifique usuelle sur Internet
- `//<user>:<password>@<host>:<port>/<url-path>`

Chemin d'accès:

- `<url-path>` dépend du "schéma",
- Avec HTTP: chemin d'accès au fichier relatif à un point de départ connu du serveur HTTP sur la machine cible

URL

- Avec http, on peut faire suivre le "url-path" par
 - `<urlpath>?requête#fragment`
- Syntaxe de la requête:
 - `param=value1&2ndparam=value2#ancre`

Unix, Réseau: Plan

- Ce que vous allez découvrir
- Usage des réseaux
- Problèmes à résoudre
- Protocoles réseaux
- Noms et Ports
- Services de nommage
- Transferts de fichiers
- **Accès à distance**
- **Ce qu'il faut retenir**

Session Distante: telnet

- telnet `[[hôte] port]`

```
neon --> telnet fluor
```

```
Trying 134.157.168.9...
```

```
Connected to fluor.pps.jussieu.fr.
```

```
Escape character is '^]'.  
login: armand
```

```
Password: ****
```

```
fluor-->
```

Session Distant: Telnet

- En utilisant le 2^{ème} argument, on peut se "connecter" à une autre application:

```
neon--> telnet fluor 7
```

```
.....
```

```
aaaa
```

```
aaaa
```

```
CTRL ]
```

```
telnet> quit
```

```
Connection closed.
```

Session Distante: Telnet

- Commandes:

- help : liste des commandes reconnues
- open *hôte* [*port*] : demande de connexion à un service distant.
- close : fermeture de connexion
- quit : idem + terminaison de telnet
- mode line : écho local, envoi ligne par ligne
character : envoi caractère par caractère
- status : état de la communication
- ! : échappement vers le shell local

Session Distante: Telnet

- Configurable au moyen du fichier:
 - ~/.telnetrc

Session Distante : ssh

- ssh version sécurisée de rsh
 - Exécute un shell interactif sur une machine distante via une connexion sécurisée
 - Comme telnet, rsh ou rlogin, nécessite d'être un utilisateur connu sur la machine distante.
- ssh [-l *nom*] *hôte* [*commande*]
 - Si -l *nom* n'est pas précisé, la demande de connexion se fait pour l'utilisateur courant.
 - Si *commande* n'est pas précisé, un shell interactif est exécuté, sinon seul *commande* est exécuté.

Session Distante : ssh

- Connexion autorisée immédiatement si :
 - machine initiale apparaît dans
 - `/etc/hosts.equiv` ou `/etc/ssh/shosts.equiv`
 - Nom utilisateur identique des deux côtés
- Ou bien
 - machine initiale et utilisateur apparaissent dans
 - `$HOME/.rhosts` ou `$HOME/.shosts` sur machine cible
 - Déconseillé, voire inhibé par administration système

Session Distante : ssh

- On peut aussi indiquer les machines connues (en fait leur clé de cryptage)
 - /etc/ssh/ssh_known_hosts
 - \$HOME/.ssh/ssh_known_hosts
- Configuration possible
 - Paramètres d'exécution de ssh
 - /etc/ssh/ssh_config, \$HOME/.ssh/ssh_config
 - Paramètres du shell lancé par ssh sur la cible
 - /etc/ssh/sshr, \$HOME/.ssh/rc

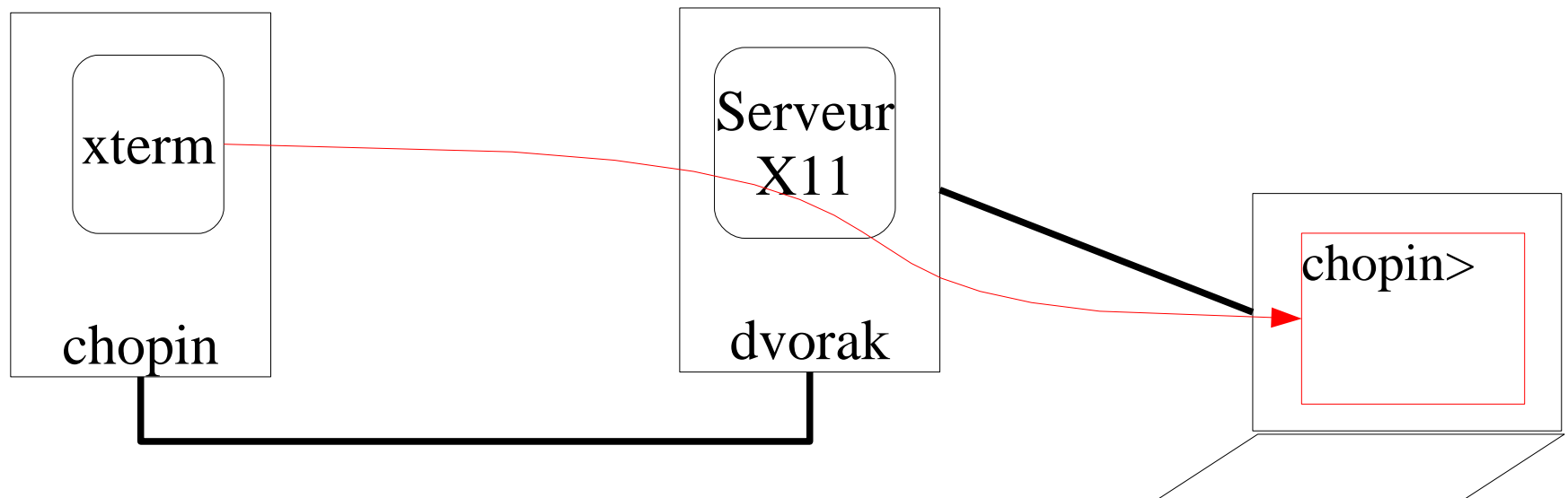
Session Distante : ssh

- Autoriser les applications graphiques (X) à ouvrir des fenêtres sur la machine locale via la connexion ssh

```
ssh [-l nom] -X hôte
```

Environnement graphique

- Afficher une fenêtre sur une machine distante
 - Variable d'environnement \$DISPLAY:
 - Ex: DISPLAY=dvorak:27.0; xterm



Environnement graphique

- Autoriser une machine distante à ouvrir une fenêtre graphique localement
 - xhost [[+/-]hôte ...]
- Permet
 - d'afficher la liste des hôtes autorisés (pas argument)
 - d'autoriser un hôte, plusieurs ou tous
 - +toto, +toto titi, +
 - d'interdire un hôte, plusieurs ou tous
 - -toto, -toto titi, -

Unix, Réseau: Plan

- Ce que vous allez découvrir
- Usage des réseaux
- Problèmes à résoudre
- Protocoles réseaux
- Noms et Ports
- Services de nommage
- Transferts de fichiers
- Accès à distance
- **Ce qu'il faut retenir**

Ce qu'il faut retenir

- Problèmes résolus par les logiciels réseaux
- Nommage des machines, nommage des applications
 - Administration et Résolution des noms de machines
- Transfert de fichiers:
 - ftp, scp, rsync
- Session à distance
 - telnet, ssh