

Introduction aux réseaux

Valeur d'accueil et de reconversion en informatique (VARI1)

Daniel Porumbel (dp.cnam@gmail.com)

<http://cedric.cnam.fr/~porumbed/var1/>

- 1 La base de l'Internet : la pile TCP/IP
- 2 Quelques problématiques de sécurité

Les adresses IP

Toute machine connectée possède une adresse IP

- exemples : 10.0.0.1, 163.173.0.1, ou 202.2.1.15
- c'est comme le numéro d'appel d'un appareil téléphonique
 - on peut avoir plusieurs IPs si on a plusieurs cartes réseaux (penser à un téléphone dual-sim)

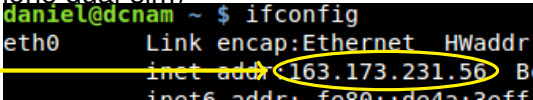
● `ifconfig` : afficher notre adresse IP

```
daniel@dcnam ~ $ ifconfig
eth0      Link encap:Ethernet  HWaddr d
          inet addr:163.173.231.56  Bca
          inet6 addr: fe80::d0e1a:2eff:
```

- `route -n` \Rightarrow afficher l'IP de la passerelle par défaut
 - ex., une box ADSL qui nous connecte à l'Internet
- `ping 163.173.228.2` \Rightarrow tester si la machine d'adresse IP 163.173.228.2 est active
- `traceroute google.fr` \Rightarrow voir les machines/routeurs qui nous relie à la machine `google.fr` (en Californie)
 - Un message est transféré de routeur en routeur (une dizaine de routeurs sont traversés) pour l'acheminer à `google.fr`

Les adresses IP

Toute machine connectée possède une **adresse IP**

- exemples : 10.0.0.1, 163.173.0.1, ou 202.2.1.15
- c'est comme le numéro d'appel d'un appareil téléphonique
 - on peut avoir plusieurs IPs si on a plusieurs cartes réseaux (penser à un téléphone dual-sim)
- `ifconfig` : afficher notre adresse IP 
- `route -n` \implies afficher l'IP de la passerelle par défaut
 - ex., une box ADSL qui nous connecte à l'Internet
- `ping 163.173.228.2` \implies tester si la machine d'adresse IP 163.173.228.2 est active
- `traceroute google.fr` \implies voir les machines/routeurs qui nous relie à la machine `google.fr`¹ (en Californie)
 - Un message est transféré de routeur en routeur (une dizaine de routeurs sont traversés) pour l'acheminer à `google.fr`

1. Utiliser `www.iplocation.net` pour voir la location géographique des routeurs.

Commandes Réseau Linux

La commande `ifconfig` affiche ou **modifie** l'IP

- `sudo ifconfig eth0 10.0.0.1/8` \implies **nouvelle IP**²
 - Si le réseau utilise un service/serveur DHCP (Dynamic Host Configuration Protocol), l'IP est configuré automatiquement (ou avec `dhclient`), et non pas avec `ifconfig`
- La machine a aussi une adresse de **diffusion** (en. : *broadcast*) qui désigne toutes les machines du (sous-)réseau

`ssh SERVEUR` : connexion au SERVEUR en mode text

- L'option `-X` de `ssh` permet d'afficher les commandes graphiques sur le serveur d'affichage (X) de la machine locale
- lancer `xeyes -display :0` en boucle `for`

2. Il faut parfois taper “`stop network-manager`” pour arrêter le gestionnaire graphique de réseaux, sinon il peut annuler l'effet d'`ifconfig`.

Commandes Réseau Linux

La commande `ifconfig` affiche ou modifie l'IP

- `sudo ifconfig eth0 10.0.0.1/8` \implies nouvelle IP²
 - Si le réseau utilise un service/serveur DHCP (Dynamic Host Configuration Protocol), l'IP est configuré automatiquement (ou avec `dhclient`), et non pas avec `ifconfig`
 - La machine a aussi une adresse de diffusion (en. : *broadcast*) qui désigne toutes les machines du (sous-)réseau

`ssh SERVEUR` : connexion au SERVEUR en mode text

- L'option `-X` de `ssh` permet d'afficher les commandes graphiques sur le serveur d'affichage (X) de la machine locale
- lancer `xeyes -display :0` en boucle `for`

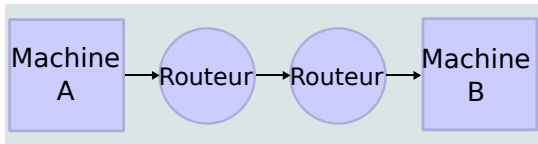
2. Il faut parfois taper "`stop network-manager`" pour arrêter le gestionnaire graphique de réseaux, sinon il peut annuler l'effet d'`ifconfig`.

La pile TCP/IP

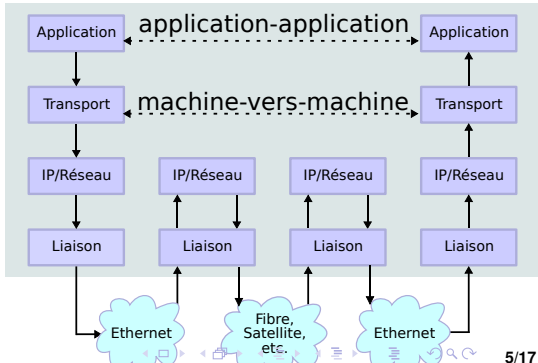
Quatre couches de protocoles qui s'appuient sur la couche physique :

- 1 Couche Applications (navigateurs Web, ssh, FTP)
- 2 Couche Transport (TCP/UDP)
- 3 Couche IP/Réseaux
- 4 Couche Liaison

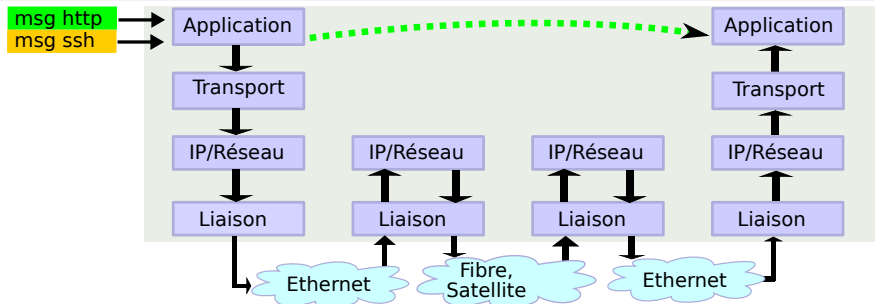
Topologie Réseau



Flux de données



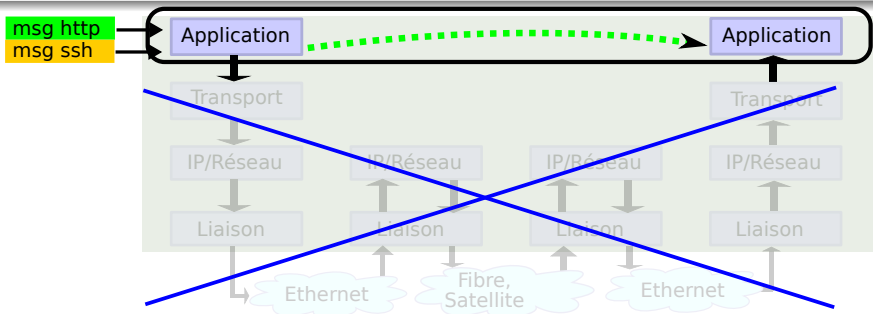
La Couche Applications



À la couche Applications on s'intéresse aux :

- programmes pour l'utilisateur (navigateur web, ssh)
- la cryptographie
- protocoles dits de haut niveau (http, https, ftp, telnet)
 - On ne s'intéresse pas aux routeurs et passerelles qui relient les machines, ni au câbles/connecteurs physiques : ces aspects sont la tâche des couches inférieures

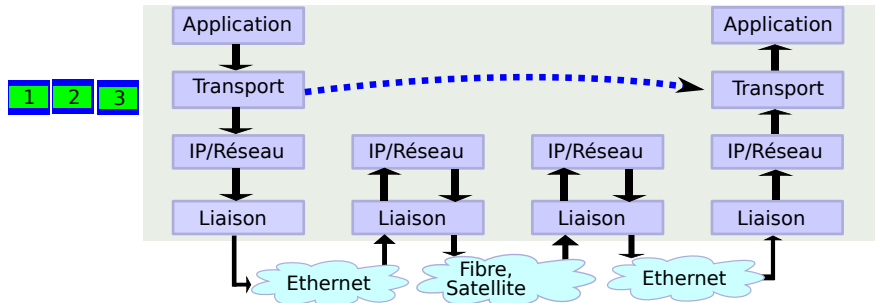
La Couche Applications



À la couche Applications on s'intéressent aux :

- programmes pour l'utilisateur (navigateur web, ssh)
- la cryptographie
- protocoles dits de haut niveau (http, https, ftp, telnet)
 - On ne s'intéresse pas aux routeurs et passerelles qui relient les machines, ni au câbles/connecteurs physiques : ces aspects sont la tâche des couches inférieures

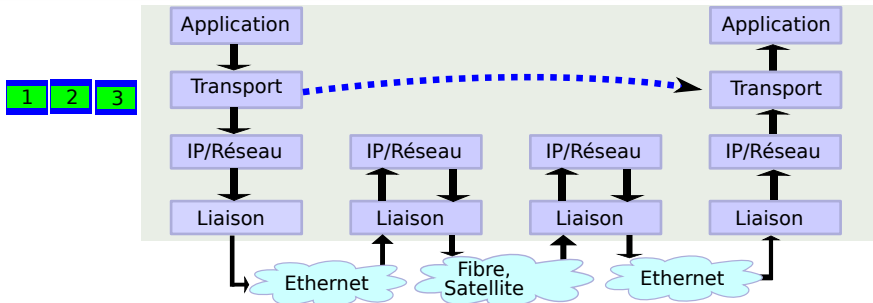
La couche Transport



- Le *message* de la couche Application est découpé en *segments* de type TCP (avec connexion) ou UDP (sans connexion)
- La communication utilise des **sockets**
 - une **socket** = une machine(IP)+ un **port**, ex. 10.0.0.1:80
 - un port \approx case courrier dans une institution (machine)
 - Exemple de connexion à `cedric.cnam.fr:80` via netcat

```
netcat cedric.cnam.fr 80<<<"get /"
```
- La transmission TCP ou UDP **ne s'intéresse pas** aux routeurs qui assurent la liaison entre les deux machines

La couche Transport

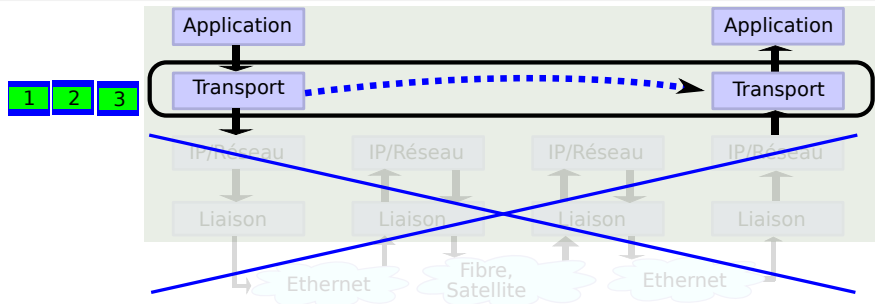


- Le *message* de la couche Application est découpé en *segments* de type TCP (avec connexion) ou UDP (sans connexion)
- La communication utilise des **sockets**
 - une **socket** = une machine(IP)+ un **port**, ex. 10.0.0.1:80
 - un port \approx case courrier dans une institution (machine)
 - Exemple de connexion à `cedric.cnam.fr:80` via netcat

```
netcat cedric.cnam.fr 80<<<"get /"
```

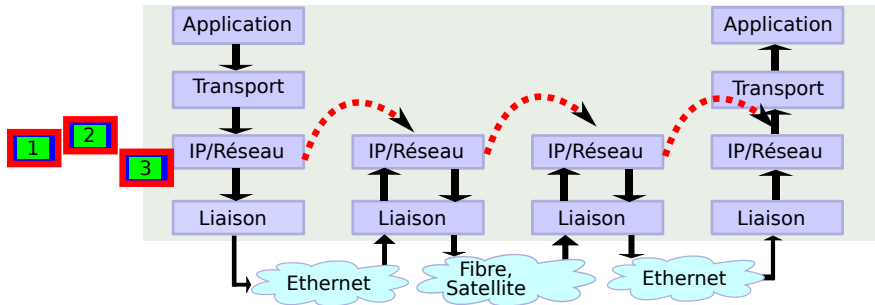
- La transmission TCP ou UDP **ne s'intéresse pas** aux routeurs qui assurent la liaison entre les deux machines

La couche Transport



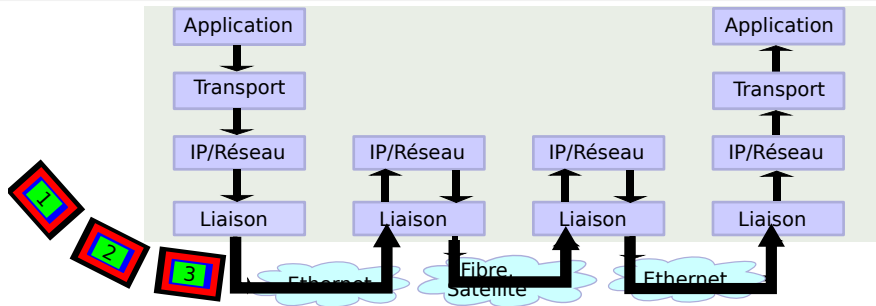
- Le *message* de la couche Application est découpé en *segments* de type TCP (avec connexion) ou UDP (sans connexion)
 - La communication utilise des **sockets**
 - une **socket** = une machine(IP)+ un **port**, ex. 10.0.0.1:80
 - un port \approx case courrier dans une institution (machine)
 - Exemple de connexion à `cedric.cnam.fr:80` via netcat
- ```
netcat cedric.cnam.fr 80<<<"get /"
```
- La transmission TCP ou UDP **ne s'intéresse pas** aux routeurs qui assurent la liaison entre les deux machines

# La couche Réseaux



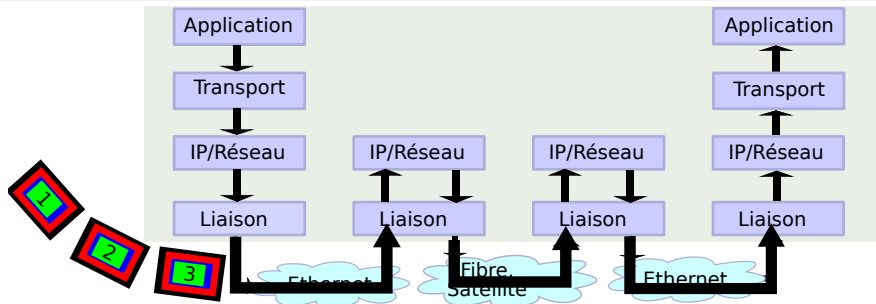
- Couche transport : segment envoyé de  $A$  vers  $B$   
 $\implies$
- Couche réseau : transfert  $A \rightarrow A_1 \rightarrow A_2 \rightarrow A_3 \dots \dots \dots \rightarrow B$ 
  - Chaque routeur peut “cacher” un pirate qui écoute la communication
- La couche *réseau* (ou couche IP) gère la circulation des *paquets* à travers le réseau en assurant leur routage.

# La couche Liaison



- La couche *liaison* gère la transmission de trames sur la couche physique
- L'en-tête d'une trame indique l'adresse **MAC** destination
  - adresse MAC=adresse physique carte réseau, c'est comme le numéro de la carte SIM (l'IP c'est le numéro d'appel)
- Le **protocole ARP** gère les associations IP  $\longleftrightarrow$  MAC.
  - Etant donné un IP de notre sous-réseau, quelle est son adresse MAC ? La commande `arp -n` affiche les MACs connus par notre machine

# La couche Liaison



- La couche *liaison* gère la transmission de trames sur la couche physique
- L'en-tête d'une trame indique l'adresse **MAC** destination
  - adresse MAC=adresse physique carte réseau, c'est comme le numéro de la carte SIM (l'IP c'est le numéro d'appel)
- Le **protocole ARP** gère les associations IP  $\longleftrightarrow$  MAC.
  - Etant donné un IP de notre sous-réseau, quelle est son adresse MAC ? La commande `arp -n` affiche les MACs connus par notre machine

# Exemple d'utilisation couche Applications

Pour aller sur `www.cnam.fr` :

- 1 Le navigateur web demande au service DNS (Domain Name Server) l'adresse IP du serveur `www.cnam.fr`
  - C'est comme si on tapait `dig www.cnam.fr`
- 2 Le DNS retourne `163.173.128.40` par exemple.
- 3 Le navigateur envoie une requête HTTP à cette adresse IP
- 4 Le serveur web répond avec un fichier `html` (+ des scripts `javascript` éventuellement)

---

Le site peut demander de déposer un **cookie** qui stocke une valeur dans le navigateur. Cette valeur peut être demandée plus tard au navigateur, ex, pour faire du pistage (publicitaire).



# Exemple d'utilisation couche Applications

Pour aller sur `www.cnam.fr` :

- 1 Le navigateur web demande au service DNS (Domain Name Server) l'adresse IP du serveur `www.cnam.fr`
  - C'est comme si on tapait `dig www.cnam.fr`
- 2 Le DNS retourne `163.173.128.40` par exemple.
- 3 Le navigateur envoie une requête HTTP à cette adresse IP
- 4 Le serveur web répond avec un fichier `html` (+ des scripts `javascript` éventuellement)
  - La communication passe par une connexion TCP sur le port `80` de `www.cnam.fr` (port  $> 1024$  pour la machine locale)

---

Le site peut demander de déposer un **cookie** qui stocke une valeur dans le navigateur. Cette valeur peut être demandée plus tard au navigateur, ex, pour faire du pistage (publicitaire).

# Exemple d'utilisation couche Applications

Pour aller sur `www.cnam.fr` :

- 1 Le navigateur web demande au service DNS (Domain Name Server) l'adresse IP du serveur `www.cnam.fr`
  - C'est comme si on tapait `dig www.cnam.fr`
- 2 Le DNS retourne `163.173.128.40` par exemple.
- 3 Le navigateur envoie une requête HTTP à cette adresse IP
- 4 Le serveur web répond avec un fichier `html` (+ des scripts `javascript` éventuellement)
  - La communication passe par une connexion TCP sur le port `80` de `www.cnam.fr` (port  $> 1024$  pour la machine locale)

---

Le site peut demander de déposer un **cookie** qui stocke une valeur dans le navigateur. Cette valeur peut être demandée plus tard au navigateur, ex, pour faire du pistage (publicitaire).

# Exemple d'utilisation couche Applications

Pour aller sur `www.cnam.fr` :

- 1 Le navigateur web demande au service DNS (Domain Name Server) l'adresse IP du serveur `www.cnam.fr`
  - C'est comme si on tapait `dig www.cnam.fr`
- 2 Le DNS retourne `163.173.128.40` par exemple.
- 3 Le navigateur envoie une requête HTTP à cette adresse IP
- 4 Le serveur web répond avec un fichier `html` ( $\oplus$  des scripts `javascript` éventuellement)
  - La communication passe par une connexion TCP sur le port 80 de `www.cnam.fr` (port  $>$  1024 pour la machine locale)

---

Le site peut demander de déposer un **cookie** qui stocke une valeur dans le navigateur. Cette valeur peut être demandée plus tard au navigateur, ex, pour faire du pistage (publicitaire).

# Exemple d'utilisation couche Applications

Pour aller sur `www.cnam.fr` :

- 1 Le navigateur web demande au service DNS (Domain Name Server) l'adresse IP du serveur `www.cnam.fr`
  - C'est comme si on tapait `dig www.cnam.fr`
- 2 Le DNS retourne `163.173.128.40` par exemple.
- 3 Le navigateur envoie une requête HTTP à cette adresse IP
- 4 Le serveur web répond avec un fichier `html` (⊕ des scripts `javascript` éventuellement)
  - La communication passe par une connexion TCP sur le port 80 de `www.cnam.fr` (port > 1024 pour la machine locale)



Le site peut demander de déposer un **cookie** qui stocke une valeur dans le navigateur. Cette valeur peut être demandée plus tard au navigateur, ex, pour faire du pistage (publicitaire).