Introduction TD-TP RLI

- Philippe.Pegon@crc.u-strasbg.fr
- Christophe.Saillard@crc.u-strasbg.fr

Introduction TD-TP RLI

- Objectif
  - Savoir répondre à cette question :
    - Comment font 2 ordinateurs en réseau pour communiquer ?
- Organisation des TP-TD
  - TD : Approfondissement des notions vues en cours et préparation des TP
  - TP : Mise en pratique

Introduction TD-TP RLI

- Evaluation
  - Un examen écrit lors du TD de la semaine du 20/03
  - Un TP noté, date communiquée une semaine avant
  - Prise en compte de la présence en TP

Introduction TD-TP RLI

Planning Prévisionnel TD/TP

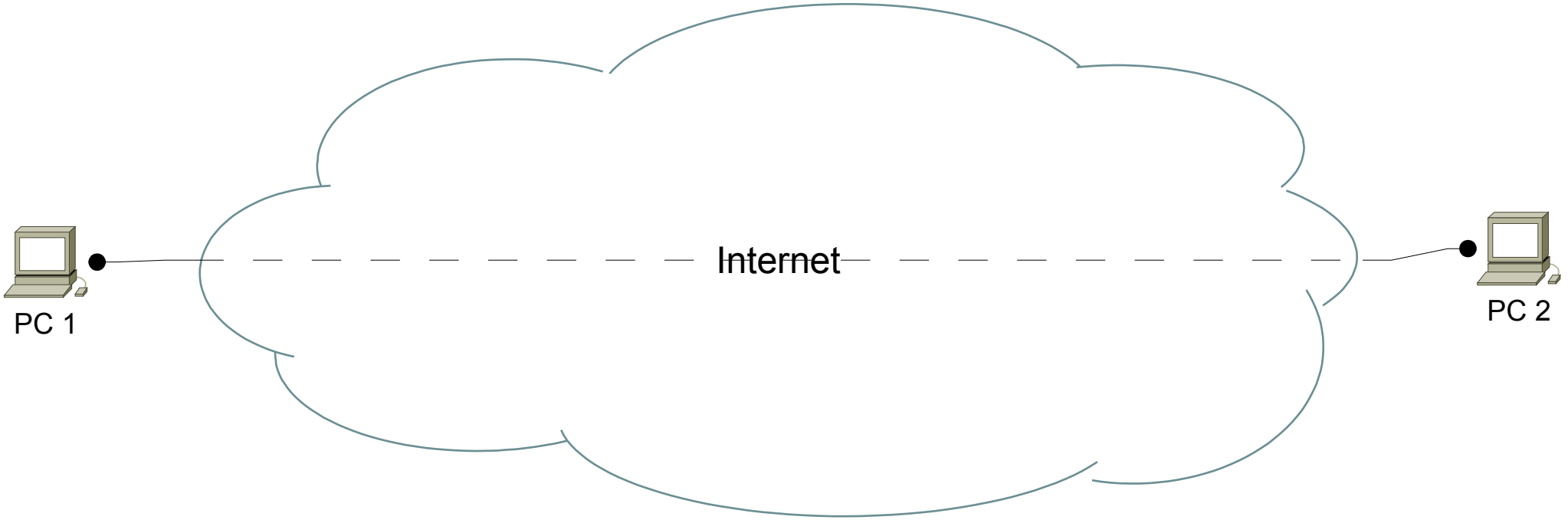
Semaine	TP	TD
27/02/06		- Intro (planning, évaluations) - Préparation TP1/TP2
06/03/06	**TP 1 :** Prise en main Cisco	- CSMA/CD - Bridging / STP
13/03/06	**TP 2 :** Prise en main Windows / Linux	- Modèle en couche (CIDR, classes d'adresse, plan d'adressage etc.)
20/03/06	**TP 3 :** Bridging / STP	Exam
27/03/06	**TP 4 :** Capture, Analyse protocoles et entêtes (IP/ARP/ICMP)	- Routage / Routeur
03/04/06	**TP 5 :** Routage statique (traceroute)	- Routage dynamique
10/04/06	**TP 6 :** Routage dynamique	- Ipv6
Vacances		
24/04/06	**TP 7 :** Ipv6 (autoconf, capture ICMP)	Prépa partiel / Visite salle machine

TP1 et TP2

- TP1 : Prise en main d'un routeur Cisco
- TP2 : Réseau sous Linux et Windows
- Objectif de ces 2 TP
  - Acquérir les bases pour les TP suivants

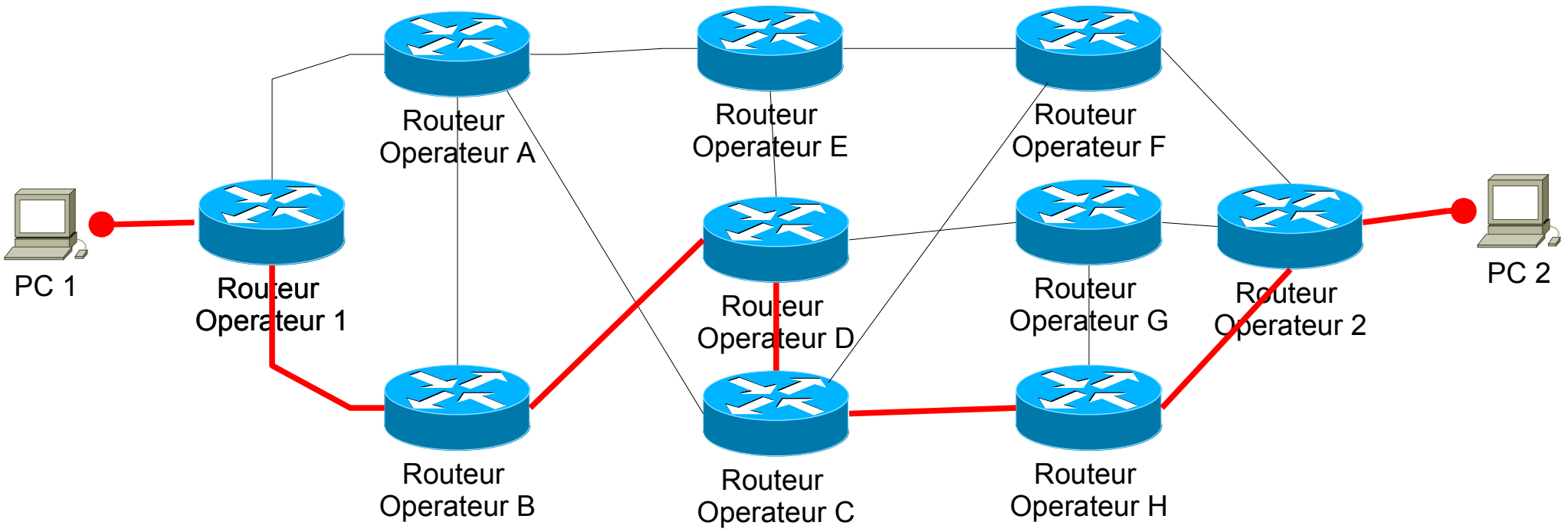
TP1 : Prise en main d'un routeur Cisco

- Un routeur à quoi ça sert ?
  - Permettre au PC1 de dialoguer avec le PC2



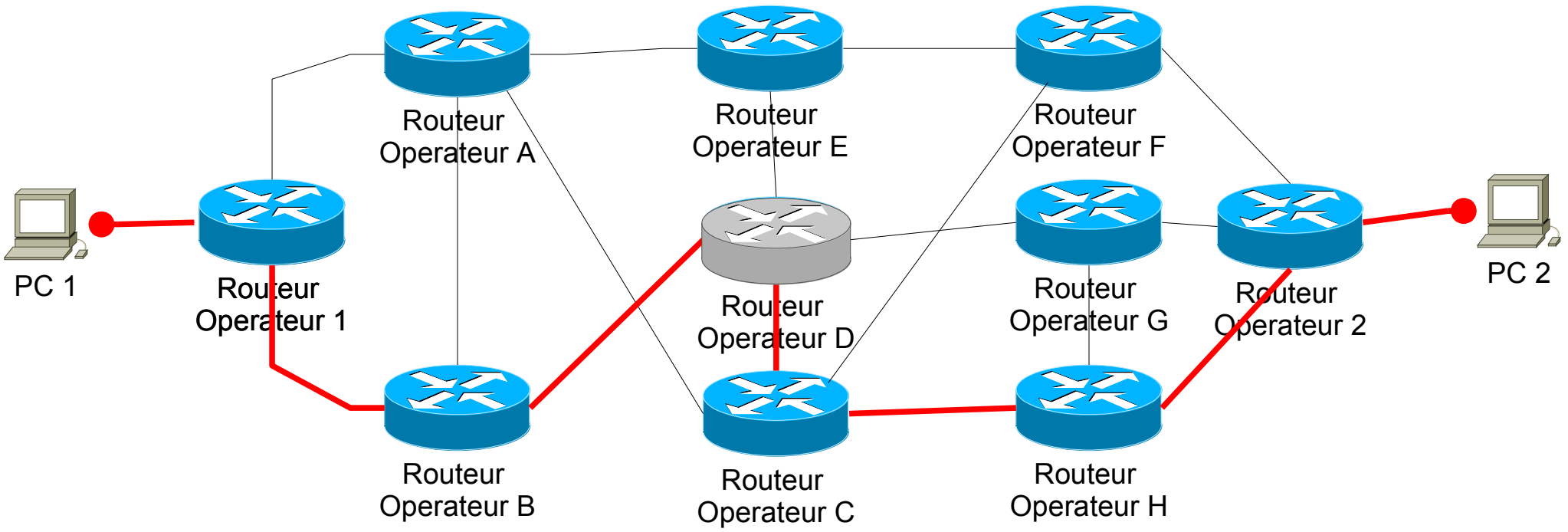
TP1 : Prise en main d'un routeur Cisco

- Un routeur à quoi ça sert ?



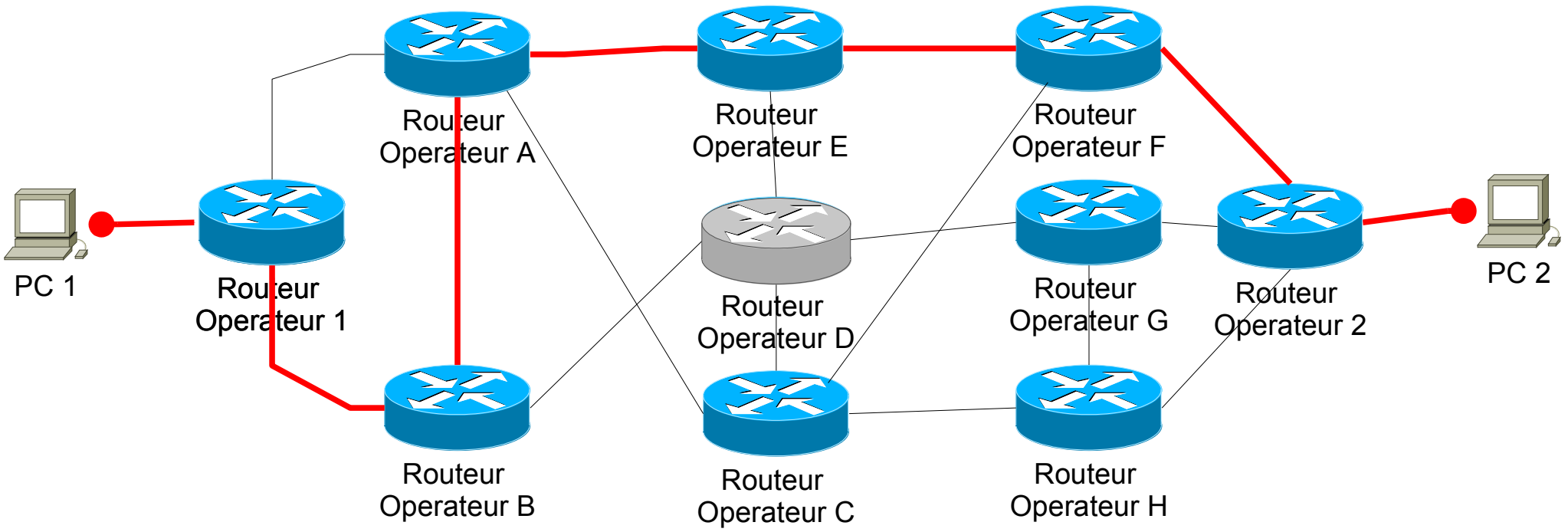
TP1 : Prise en main d'un routeur Cisco

- Un routeur à quoi ça sert ?
  - Le routeur D tombe en panne



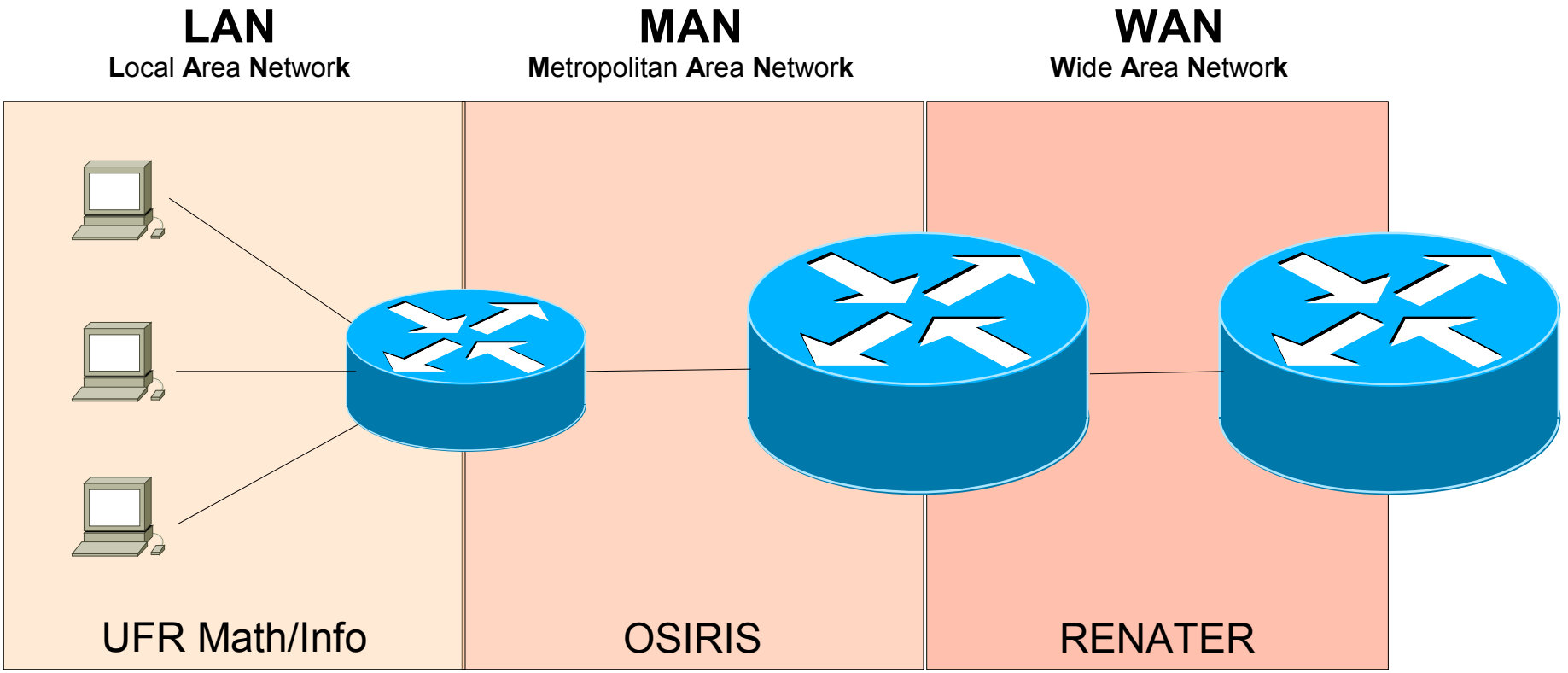
TP1 : Prise en main d'un routeur Cisco

- Un routeur à quoi ça sert ?
  - Le chemin est recalculé



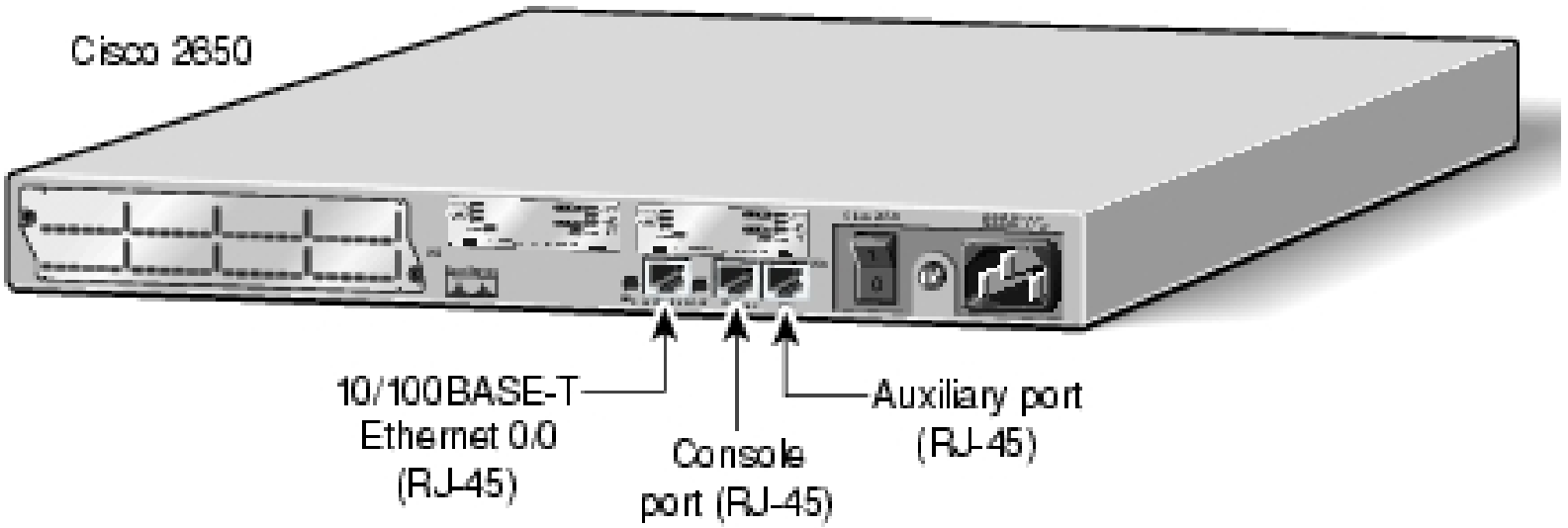
TP1 : Prise en main d'un routeur Cisco

- Un routeur, plusieurs interfaces



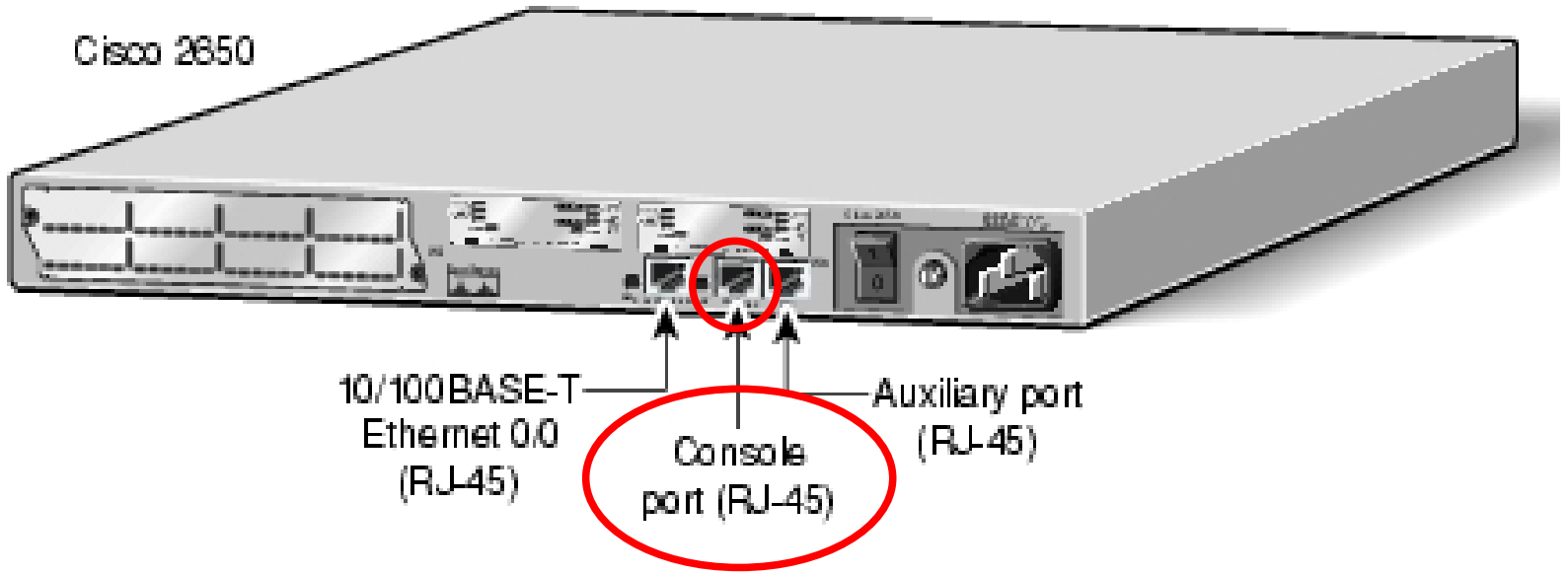
TP1 : Prise en main d'un routeur Cisco

- Les routeurs de TP
  - Cisco 2600



TP1 : Prise en main d'un routeur Cisco

- Généralités sur les équipements Cisco
  - **Port Console** : accès administratif local à partir d'un terminal ASCII ou d'une émulation de terminal (**Hyperterminal** sous Windows, **Minicom** sous Unix.) A relier au port série de votre ordinateur.

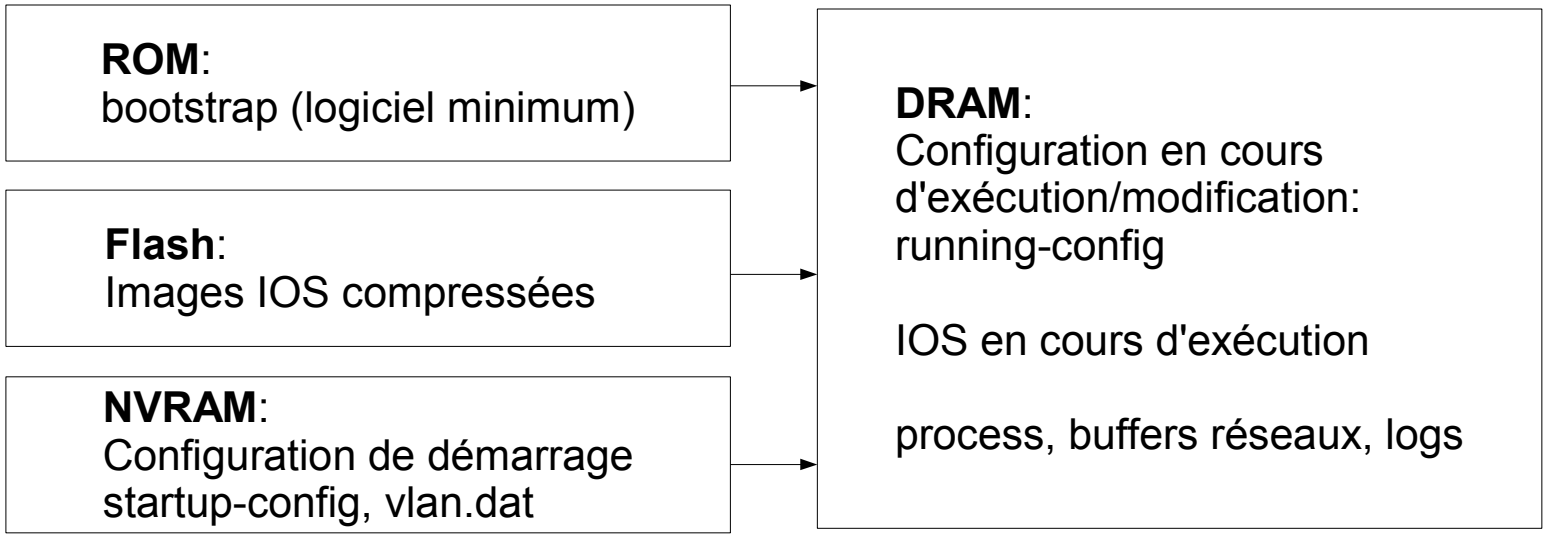


TP1 : Prise en main d'un routeur Cisco

- Généralités sur les équipements Cisco
  - Acronymes de base Cisco
    - **IOS** *(Internetwork Operating System)*, c'est l'OS de tous les équipements Cisco
    - **CLI** *(Command Line Interface),* désigne l'interface en ligne de commande.
    - Types de mémoires
      - **DRAM** (volatile)
      - **NVRAM** (non-volatile)
      - **Flash** (non-volatile)
      - **ROM** (non-volatile)

TP1 : Prise en main d'un routeur Cisco

- Généralités sur les équipements Cisco
  - Séquence d'initialisation d'un routeur
    - Chargement du boostrap de ROM vers DRAM
    - Test de la plate-forme
    - Chargement de l'IOS de Flash vers DRAM
    - Chargement de la config de NVRAM vers DRAM



TP1 : Prise en main d'un routeur Cisco

- Configurer le routeur
  - Connexion via le port console
    - Via un terminal vt100 ou émulation (Hyperterminal/Minicom)
    - Paramétrage :
      - Vitesse : 9600 bps
      - Taille : 8 bits
      - Parité : non
      - Bit d'arret : 1
      - Controle de flux : non

TP1 : Prise en main d'un routeur Cisco

- Configurer le routeur
  - Navigation et Configuration
    - Pour obtenir la liste des commandes, taper : ?
    - « ? » permet aussi de compléter une commande.

```
Switch> ping ?
WORD  Ping destination address or hostname
ip    IP echo
tag   Tag encapsulated IP echo
```

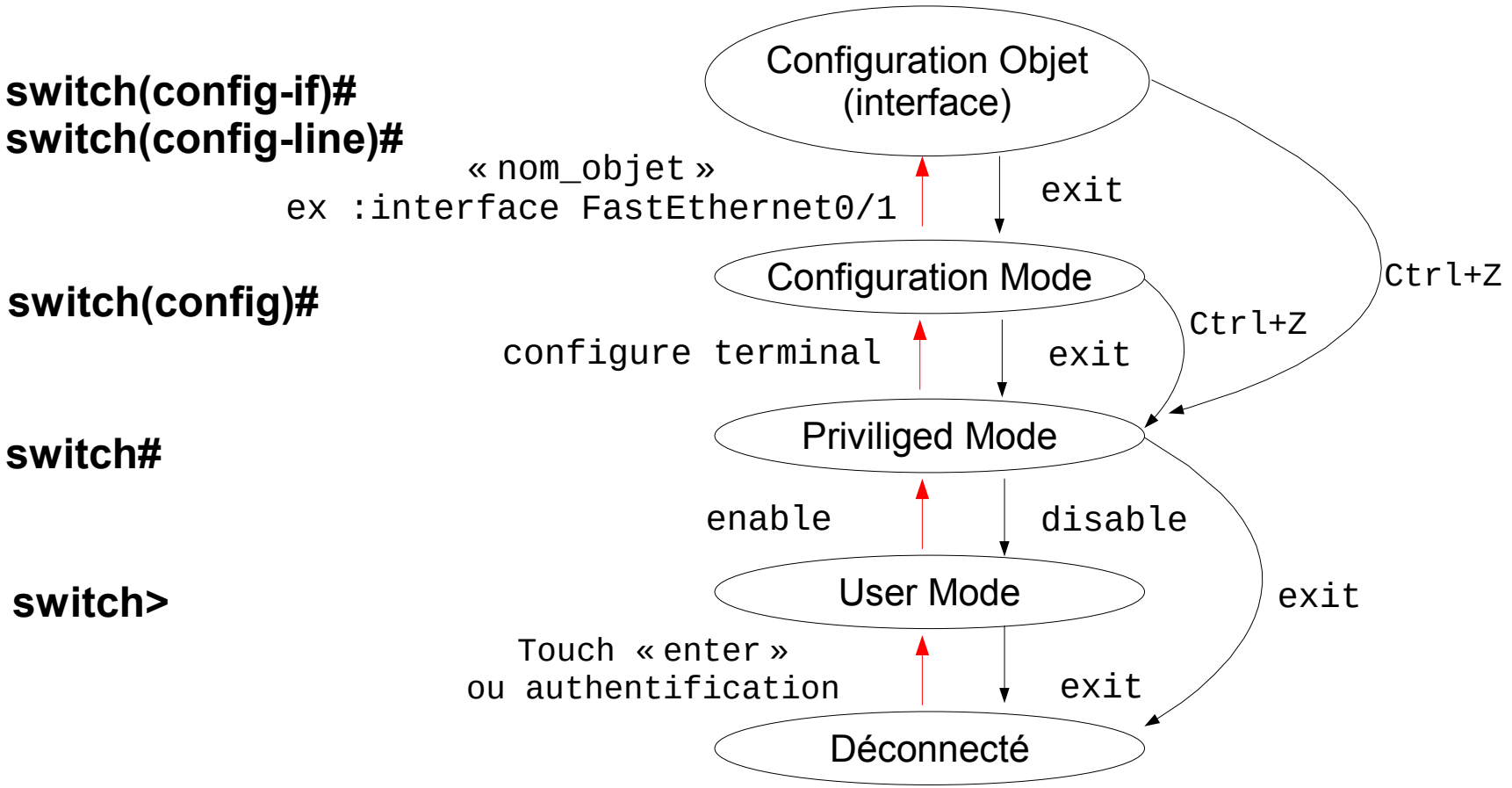
    - La « complétion » des commandes se fait avec la touche **<TAB>**.
    - Si le résultat d'une commande dépasse la taille du terminal
      - Appuyer sur **entrée** pour voir la suite **ligne par ligne**
      - Appuyer sur la **barre espace** pour voir la suite **page par page**.

TP1 : Prise en main d'un routeur Cisco

- Configurer le routeur
  - Les différents modes de configuration

Mode	Prompt visuel
Mode utilisateur	router>
Mode privilégié	router#
Mode Configuration	router(config)#
Mode Configuration Objet	router(config-if)#

TP1 : Prise en main d'un routeur Cisco



TP1 : Prise en main d'un routeur Cisco

- Configurer le routeur
  - Exemple 1 : se connecter et configurer une interface
    - Etape 1 : connexion via la port console
      - Prompt `routeur>`
    - Etape 2 : On tape « enable » puis on saisit le mdp
      - Prompt `routeur#`
    - Etape 3 : On tape « configure terminal »
      - Prompt `routeur(config)#`
    - Etape 4 : On tape « interface X »
      - Prompt `routeur(config-if)#`

TP1 : Prise en main d'un routeur Cisco

- Configurer le routeur
  - Gérer la configuration
    - Se mettre en mode enable
    - Voir la configuration active, running-config (DRAM)
      `router# show running-config`
    - Voir la configuration de démarrage, startup-config:
      `router# show startup-config`
    - Copie de la conf courante vers la conf de démarrage
      `router# copy running-config startup-config`

      > A faire après chaque modification de configuration
      > On peut également taper « write mem »

TP1 : Prise en main d'un routeur Cisco

- Configurer le routeur
  - Le système de fichier
    - Pour tout visualiser taper : `router>` **`dir all-filesystems`**
    - Copier un fichier:

      `router#` **`copy flash:/config.text flash:/config.origine`**
    - Renommer un fichier:

      `router#` **`rename flash:/<ancien_nom> flash:/<nouveau_nom>`**
    - Supprimer un fichier:

      `router#` **`delete flash:/<fichier>`**

TP1 : Prise en main d'un routeur Cisco

- Configurer le routeur
  - Pour chaque TP
    - Identifiants par défaut
      - Login : **Cisco**
      - Password : **Cisco**
      - **Important :** ne les changez surtout pas !
    - Démarrer sur une configuration vierge
      - En mode enable
        - Supprimer la config de démarrage

          ```
          routeur# erase startup-config
          ```
        - Redémarrer le routeur

          ```
          routeur# reload
          ```
          - Après le reboot répondre 'no' à l'assistant de configuration

TP1 : Prise en main d'un routeur Cisco

- Configurer le routeur
  - Définir un nom

```
routeur(config)# hostname <nom du routeur>
```

  - Ajouter une passerelle par défaut

```
routeur(config)# ip default gateway A.B.C.D
```

  - Activer la résolution de nom

```
routeur(config)# ip domain-lookup
routeur(config)# ip domain-name u-strasbg.fr
routeur(config)# ip name-server 130.79.200.1
```

TP1 : Prise en main d'un routeur Cisco

- Configurer le routeur
  - Affecter une adresse IP à une interface
    - On se déplace dans le menu de l'interface
      ```
      Router(config)# interface ethernet X
      ```
    - On spécifie l'adresse IP et le masque associé
      ```
      Router(config-if)# ip address A.B.C.D E.F.G.H
      ```
    - On active l'interface
      ```
      Router(config-if)# no shutdown
      ```
    - **Remarque générale Cisco**
      - Pour annuler une commande on tape « no » devant
      - Ex :
        - ip address A.B.C.D E.F.G.H pour affecter l'adresse
        - no ip address A.B.C.D E.F.G.H pour la supprimer

TP1 : Prise en main d'un routeur Cisco

- Configurer le routeur
  - La configuration à distance
    - Avec un serveur de terminal (relié au port console du routeur)
    - Par Telnet ou SSH
      - Authentification nécessaire
        - 1. Chiffrer les mots de passes dans la configuration
          ```
          router(config)# service password-encryption
          ```
        - 2. Définir un mot de passe pour les terminaux « virtuels » (de 0 à 4)
          ```
          router(config)# line vty 0 4
          router(config)# password <mot de passe>
          router(config)# login
          ```
        - 3. Définir un mot de passe « enable »
          ```
          router(config)# enable secret <mot de passe>
          ```
        - 4. Sur votre poste de travail : telnet <ip routeur>

TP1 : Prise en main d'un routeur Cisco

- Configurer le routeur
  - Enregistrer / charger une configuration par tftp

```
routeur# copy startup-config tftp:
```

TP1 : Prise en main d'un routeur Cisco

- Configurer le routeur
  - Diagnostiquer un problème
    - Tester la connectivité avec la commande ping
    - Consulter la configuration
      ```
      router# show running-configuration
      ```
    - Consulter les logs
      ```
      router# show logging
      ```
    - Consulter l'état d'une interface
      ```
      router# show interface X
      ```
    - Quel est l'OS du routeur, quelles sont ses caractéristiques
      ```
      router# show version
      router# show system
      ```

TP Linux / Windows

- Interfaces réseaux
  - Nommage des interfaces
  - Etat du lien physique
  - Configurer / Visualiser une adresse IP sur une interface
  - Ajouter d'un alias à une interface
- Configurer / Visualiser les routes
- Configurer le résolveur de nom (DNS)
- Consultation de la table arp
- Vérifier la connectivité vers une machine
- Tracer le chemin vers une machine
- Capturer le trafic

Linux : interfaces réseaux (1)

- Interfaces physiques nommées eth0, eth1 ...
- *ethtool* permet de visualiser l'état d'un lien
  - câble connecté / déconnecté
  - vitesse
  - duplex
  - ...

```
debi:~# ethtool eth0
Settings for eth0:
        Supported ports: [ MII ]
        Supported link modes:   10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Half 1000baseT/Full
        Supports auto-negotiation: Yes
        Advertised link modes:  10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Half 1000baseT/Full
        Advertised auto-negotiation: Yes
        Speed: 100Mb/s
        Duplex: Full
        Port: Twisted Pair
        PHYAD: 1
        Transceiver: internal
        Auto-negotiation: on
        Supports Wake-on: g
        Wake-on: d
        Current message level: 0x000000ff (255)
        Link detected: yes
```

Linux : interfaces réseaux (2)

- Commande *ifconfig*
  - pour visualiser

```
debi:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:16:3E:7F:21:6D
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0 MiB)  TX bytes:0 (0 MiB)
```

  - pour configurer un adresse IP

```
debi:~# ifconfig eth0 192.168.1.1 netmask 255.255.255.0
debi:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:16:3E:7F:21:6D
          inet addr:192.168.1.2  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:61488 errors:0 dropped:0 overruns:0 frame:0
          TX packets:54819 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:36743162 (35.0 MiB)  TX bytes:19210934 (18.3 MiB)
```

Linux : interfaces réseaux (3)

- Interface *lo* spéciale (loopback)
  - toujours présente
  - nécessaire pour les communications internes de l'OS
  - adresse IP de loopback : 127.0.0.1

```
debi:~# ifconfig lo
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:33301 errors:0 dropped:0 overruns:0 frame:0
          TX packets:33301 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:15973077 (15.2 MiB)  TX bytes:15973077 (15.2 MiB)
```

Linux : interfaces réseaux (4)

- alias sur une interface
- permet d'avoir plusieurs adresses IP sur une même interface physique

```
debi:~# ifconfig eth0:1 192.168.1.3 netmask 255.255.255.0
debi:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:16:3E:7F:21:6D
          inet addr:192.168.1.2  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:79152 errors:0 dropped:0 overruns:0 frame:0
          TX packets:72733 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:38490380 (36.7 MiB)  TX bytes:24477322 (23.3 MiB)

eth0:1    Link encap:Ethernet  HWaddr 00:16:3E:7F:21:6D
          inet addr:192.168.1.3  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:38970 errors:0 dropped:0 overruns:0 frame:0
          TX packets:38970 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:19033725 (18.1 MiB)  TX bytes:19033725 (18.1 MiB)
```

Windows : interfaces réseaux

- En mode texte
  - *ipconfig*
    - visualiser l'adresse IP
    - visualiser l'état du lien
  - *netsh*
    - ligne de commande (cli)
    - permet de faire toutes les opérations relatives au réseau
    - traduction en français le rend difficilement utilisable
    - peu intuitif
- En mode graphique
  - vous savez tous faire...

Linux : routes

- Commande *route*
  - permet d'ajouter une route

```
debi:~# route add -net 0.0.0.0 netmask 0.0.0.0 gw 192.168.1.1
```

  - ou équivalent (pour la route par défaut)

```
debi:~# route add default gw 192.168.1.1
```

  - permet de visualiser les routes

```
debi:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.1.0     *               255.255.255.0   U     0      0        0 eth0
default         192.168.1.1     0.0.0.0         UG    0      0        0 eth0
```

  - route sous Windows

Linux : fichier de configuration

- Ces modifications ne sont pas gardées après reboot
- Chaque distribution Linux a des fichiers de configuration différents
- Sous Debian

```
debi:~# cat /etc/network/interfaces
# Used by ifup(8) and ifdown(8). See the interfaces(5) manpage
or
# /usr/share/doc/ifupdown/examples for more information.
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
    address 192.168.1.2
    network 192.168.1.0
    netmask 255.255.255.0
    broadcast 192.168.1.255
    gateway 192.168.1.1
```

Linux : configuration DNS

- Permet de résoudre non -> adresse IP (et l'inverse)
- Fichier */etc/resolv.conf*

```
debi:~# cat /etc/resolv.conf
search u-strasbg.fr
nameserver 192.168.1.1
```

- Commande *nslookup*

```
debi:~# nslookup www.google.fr
Server:         192.168.1.1
Address:        192.168.1.1#53

Non-authoritative answer:
www.google.fr   canonical name = www.google.com.
Name:   www.google.com
Address: 66.249.93.104
```

Linux / Windows : table arp

- Commande *arp* sous Linux et Windows
  - permet de consulter la table arp

```
debi:~# arp -a
? (192.168.1.1) at 00:16:3E:3B:93:ED [ether] on eth0
```

  - permet de « nettoyer » la table arp

```
debi:~# arp -d 192.168.0.1
debi:~# arp -a
debi:~#
```

Linux / Windows : connectivité

- Commande *ping* sous Linux et Windows
  - permet de vérifier la connectivité d'une machine
  - visualiser les délais de transmission

```
debi:~# ping -c 5 192.168.0.101
PING 192.168.0.101 (192.168.0.101) 56(84) bytes of data.
64 bytes from 192.168.0.101: icmp_seq=1 ttl=63 time=1.72 ms
64 bytes from 192.168.0.101: icmp_seq=2 ttl=63 time=0.771 ms
64 bytes from 192.168.0.101: icmp_seq=3 ttl=63 time=0.758 ms
64 bytes from 192.168.0.101: icmp_seq=4 ttl=63 time=0.804 ms
64 bytes from 192.168.0.101: icmp_seq=5 ttl=63 time=0.861 ms

--- 192.168.0.101 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4011ms
rtt min/avg/max/mdev = 0.758/0.984/1.728/0.374 ms
```

Linux / Windows : traceroute

- Commande *traceroute* sous Linux
  - permet tracer le chemin vers une machine
  - visualiser les délais vers chaque saut

```
debi:~# traceroute -n 192.168.0.101
traceroute to 192.168.0.101 (192.168.0.101), 30 hops max, 38 byte
packets
 1  192.168.1.1  1.200 ms  0.936 ms  0.690 ms
 2  192.168.0.101  1.360 ms  1.096 ms  1.317 ms
```

- Commande *tracert* sous Windows
  - équivalente au *traceroute* de Linux

Linux : capture du trafic (1)

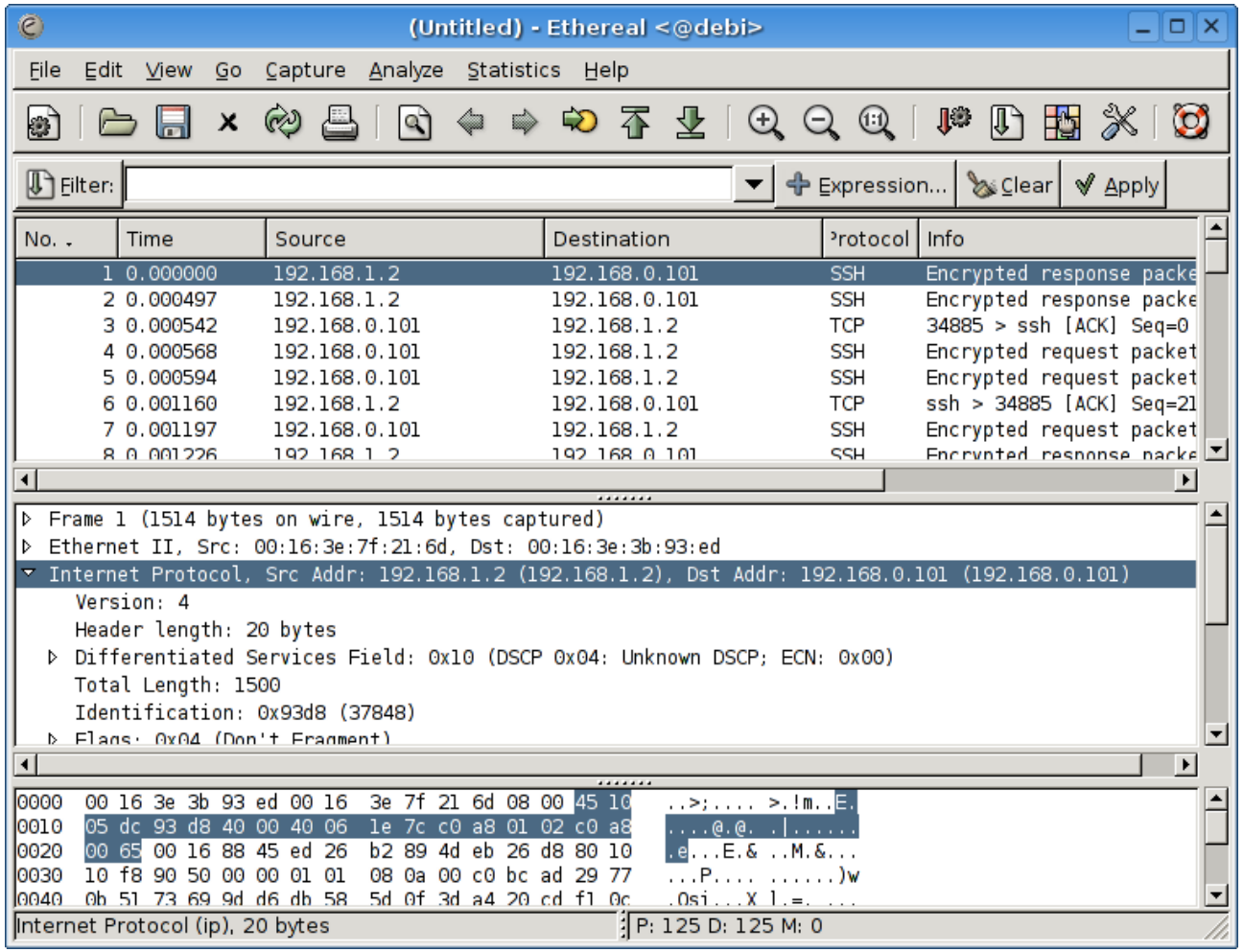
- la libcap permet de capturer du trafic
- en mode texte : *tcpdump*

```
debi:~# tcpdump -ni eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
00:19:52.639172 IP 192.168.1.2.22 > 192.168.0.101.33120: P 3481995539:3481995651(112) ack
796527980 win 2896 <nop,nop,timestamp 12616364 695519085>
00:19:52.641078 IP 192.168.1.2.22 > 192.168.0.101.33120: P 112:224(112) ack 1 win 2896
<nop,nop,timestamp 12616364 695519085>
...

70 packets captured
92 packets received by filter
0 packets dropped by kernel
```

Linux : capture du trafic (2)

- en mode graphique : *ethereal*



Windows : capture du trafic

- la lib winpcap permet de capturer du trafic
- *windump* permet de faire « l'équivalent » de *tcpdump* sous Linux
  - http://www.winpcap.org/windump/