Commandes réseaux sous Linux (rédigé pour AlmaLinux Server)

Hainaut Patrick 2022

But de cette présentation

- Apprendre les commandes réseaux de base

Introduction

- Nous allons voir ici les commandes de base utiles à la manipulation des réseaux
- Certaines de ces commandes seront revues dans d'autres présentations mais un rappel ne fait jamais de tort ;-)
- Comme pour les commandes se rapportant aux fichiers et répertoires, nous ne verrons pas toutes les options possibles de chaque commande, l'aide est là pour ça …

Noms des cartes réseau sous Linux

- Depuis la sortie du noyau Linux 4.4, les cartes réseau sont nommées en fonction de leur nature et de leur emplacement dans l'ordinateur:
  - D'abord un préfixe en deux lettres:
    - en pour Ethernet
    - wl pour Wireless
    - …
  - Une lettre suivant le type de bus:
    - o pour On board (carte mère)
    - s pour un slot PCI Express
    - p pour un slot PCI
    - ...
  - Un numéro de port
- Exemple: eno1

Noms des cartes réseau sous Linux

- Pour les cartes enfichées, on peut trouver en plus: s<slot>[f<function>][d<dev_port>]
- Pour les cartes USB, on peut trouver en plus: s<slot>[f<function>][u<port>][..][c<config>][i<interface>]

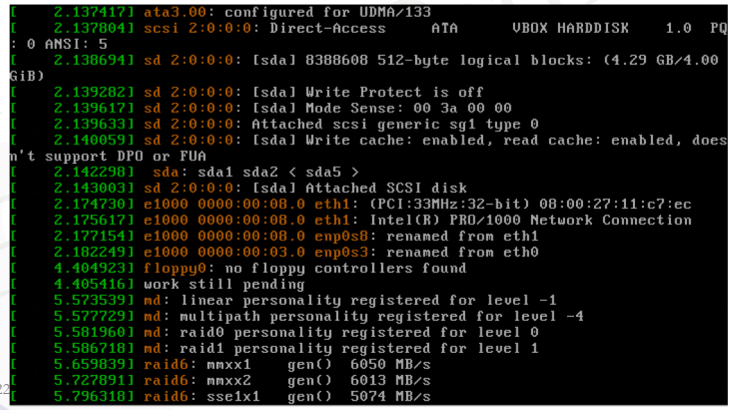
- Ce qui peut donner: enp5s3 (carte PCI Express)
  wlp3s1 (carte WIFI)
  enp2s0f1 (2ème port Ethernet d'une carte PCI Express avec 2 ports)
  …
- Dans le reste de la présentation, nous utiliserons les cartes réseau enp0s3 (équivalent eth0) et enp0s8 (équivalent eth1), adaptez les commandes en fonction de vos cartes réseau ;-)

Noms des cartes réseau sous Linux

- Pourquoi ce système de noms prévisibles (Predictable Network Interface Names) ?
  - Parce qu'avec l'ancien système de nommage eth0, eth1, …, les interfaces étaient nommées suivant l'ordre de détection par le système
  - Qu'après un redémarrage ou un changement de carte réseau, le nom de la carte pouvait donc avoir changé, ce qui posait problème pour tous les services réseaux basés sur le nom de la carte
- Cependant, ce système étant relativement nouveau, vous trouverez encore beaucoup de littérature technique employant les noms ethX
- Rien ne change, il suffit d'adapter le nom

dmesg: Visualisation du processus de démarrage

- dmesg permet de revoir tout le processus de démarrage de l'OS et de voir ainsi tout le matériel détecté; des disques durs aux cartes réseaux (utile pour connaître leur nombre et leur nom)



ifconfig: Visualisation des cartes réseau

- ifconfig permet de voir les cartes réseaux et leur configuration
- ifconfig seul n'affiche que les interfaces configurées
- ifconfig -a affiche toutes les interfaces présentes dans le système qu'elles soient actives ou pas
- ifconfig enp0s3 affiche les informations relatives à cette interface
- ifconfig --help ou man ifconfig pour voir toutes les options

ifconfig: Configuration des cartes réseau

- ifconfig *interface ip_address netmask* assigne des paramètres IP (n'est effectif que durant la session, disparait au redémarrage)

  Exemple: ifconfig enp0s3 192.168.1.1 netmask 255.255.255.0

- Ifconfig *interface* hw ether *mac_address* change l'adresse MAC (disparait au redémarrage)

  Exemple: ifconfig enp0s3 hw ether 00:AF:21:2B:38:F0

ifconfig: Montage et démontage des cartes réseau

- ifconfig enp0s3 up active (monte) l'interface
- ifconfig enp0s3 down désactive (démonte) l'interface

ifconfig: Visualisation des cartes réseau

```
root@srvLX01:~# ifconfig enp0s3
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:ab:c4:43
          inet addr:192.168.1.123  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:feab:c443/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:789 errors:0 dropped:0 overruns:0 frame:0
          TX packets:39 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:118052 (118.0 KB)  TX bytes:3960 (3.9 KB)

root@srvLX01:~# _
```

- Informations affichées:
  - Link encap:Ethernet: l'interface est de type Ethernet
  - Hwaddr: adresse MAC
  - Inet adr: adresse IP
  - Bcast: adresse de diffusion
  - Mask: masque de sous-réseau

ifconfig: Visualisation des cartes réseau

```
root@srvLX01:~# ifconfig enp0s3
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:ab:c4:43
          inet addr:192.168.1.123  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:feab:c443/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:789 errors:0 dropped:0 overruns:0 frame:0
          TX packets:39 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:118052 (118.0 KB)  TX bytes:3960 (3.9 KB)

root@srvLX01:~# _
```

- Informations affichées:
  - UP: interface activée
  - BROADCAST: l'interface traite la diffusion
  - RUNNING: l'interface est connectée
  - MULTICAST: l'interface traite la multi-diffusion

ifconfig: Visualisation des cartes réseau

```
root@srvLX01:~# ifconfig enp0s3
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:ab:c4:43
          inet addr:192.168.1.123  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:feab:c443/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:789 errors:0 dropped:0 overruns:0 frame:0
          TX packets:39 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:118052 (118.0 KB)  TX bytes:3960 (3.9 KB)

root@srvLX01:~# _
```

- Informations affichées:
  - MTU (Maximum Transmission Unit): C'est la plus grande longueur de trame en octets sans fragmentation permise sur l'interface
  - Metric: coût d'un chemin pour le protocole de routage dynamique RIP

ifconfig: Visualisation des cartes réseau

```
root@srvLX01:~# ifconfig enp0s3
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:ab:c4:43
          inet addr:192.168.1.123  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:feab:c443/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:789 errors:0 dropped:0 overruns:0 frame:0
          TX packets:39 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:118052 (118.0 KB)  TX bytes:3960 (3.9 KB)

root@srvLX01:~# _
```

- Informations affichées:
  - RX packets: nombre de paquets reçus
  - TX packets: nombre de paquets envoyés
    - Errors (paquets erronés) , dropped (paquets abandonnés), overruns (paquets engorgés), frame (Nbr de trames erronées -> bad CRC), carrier (problème de signal), collisions (paquets en collision -> half duplex), txqueuelen (longueur de la file d'attente en émission)

    -> Indicateurs de santé de l'interface et du réseau

ifconfig: Visualisation des cartes réseau

```
root@srvLX01:~# ifconfig enp0s3
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:ab:c4:43
          inet addr:192.168.1.123  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:feab:c443/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:789 errors:0 dropped:0 overruns:0 frame:0
          TX packets:39 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:118052 (118.0 KB)  TX bytes:3960 (3.9 KB)

root@srvLX01:~# _
```

- Informations affichées:
  - RX bytes: nombre de paquets reçus, en bytes
  - TX bytes: nombre de paquets envoyés, en bytes

ifconfig: Interface loopback (lo)

- C'est une interface virtuelle présente au niveau interne de l'ordinateur
- C'est l'interface logique de boucle locale nommée localhost
- Elle permet de faire tourner les services réseaux sans nécessiter la présence d'une interface physique
- Elle est requise et obligatoire pour certains services (Ex.: Samba)
- En IPv4, localhost correspond à une adresse IP de la plage 127.0.0.0/8, habituellement à l'adresse 127.0.0.1
- En IPv6, localhost correspond à l'adresse ::1

ifconfig: Exercices

- Comment surveiller en temps réel l'activité réseau sur enp0s3 ?
- Soit un PC configuré avec 4 cartes réseaux identiques et de même marque. Comment pouvoir facilement identifier physiquement chacune d'entre elles ?

ifconfig: Atelier 1

- Relevez les paramètres IP de votre machine
- Démontez les 2 interfaces lo et enp0s3 (ou équivalent)
- Entrez les commandes suivantes. Que constatez-vous ?

  # ping localhost
  # ping @IP_du_pc
  # telnet 127.0.0.1
- Remontez les interfaces lo et enp0s3 et vérifiez leur fonctionnement.
- Relevez, dans un tableau croisé, les couples RX/TX de ces 2 interfaces
- Pingez le loopback. Que constatez-vous ?
- Pingez l'@IP_du_pc. Que constatez-vous ?
- Pingez l'@IP_du_pc_du_voisin. Que constatez-vous ?

ifconfig: Atelier 2

- Relevez le couple @IP/Masque de enp0s3
- Changer les paramètres IP de enp0s3 pour qu'elle devienne la $x_{ème}$ interface du réseau 10.104.0.0/16 et que son @IP soit unique sur ce réseau
- Pingez le serveur 10.103.0.1. Que constatez-vous ?
- Pingez la machine de votre voisin. Que constatez-vous ?
- Reconfigurez enp0s3 pour retrouver les paramètres IP d'origine

ifconfig: Atelier 3

- Relevez l'@MAC de enp0s3
- Changez-la en 00:01:02:03:04:0x (avec x un numéro unique pour éviter les duplications d'adresses MAC sur le réseau !)
- Relevez l'@MAC de enp0s3
- Reconfigurez enp0s3 pour retrouver l'@MAC d'origine
- Utilités ?

Complément: le MTU : définition

- Lors d'une transmission de données informatiques, le MTU est la taille maximale d'un datagramme pouvant être transmis en une seule fois (sans fragmentation) sur une interface
- Chaque transmission de trame est définie par le corps (=MSS=maximum segment size) qui défini le plus grand segment d'informations TCP pouvant être transmis, et l'entête (header en anglais)
- Soit MTU=MSS + TCP/IP headers

Complément: le MTU : valeurs théoriques

- Quelques valeurs de MTU maximum:
  - Ethernet: 1500 octets par défaut
  - Connexions en PPPoE (ADSL sur RJ45 par exemple): 1492 octets PPPoE rajoute une couche de 8 octets, ce qui donne 1500 - 8 = 1492
  - Connexions bas dédit (PSTN): 576 octets
- Quelques valeurs de MTU minimum:
  - IPv4: 68 octets
  - IPv6: 1280 octets

Complément: le MTU : commandes

- Trouver le MTU sous Linux:

  ifconfig *interface* | grep -Eoi "MTU:[0-9]+"

  Exemple: ifconfig enp0s3 | grep -Eoi "MTU:[0-9]+"

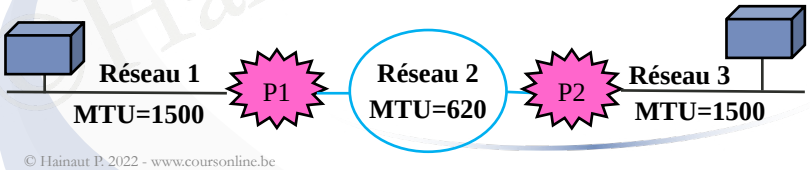
- Changer le MTU sous Linux:

  ifconfig *interface* mtu *valeur_mtu*

  Exemple: ifconfig enp0s3 mtu 1400

Complément: le MTU : PMTU

- On parle de *Path MTU* pour désigner la taille maximale entre une machine source et une machine destination
- Il correspond au plus petit MTU des interfaces par où le paquet est transite
- Le *Path MTU* peut varier dans le temps, suite à un reroutage par exemple
- Il peut également être asymétrique



Complément: le MTU : PMTUd

- Le Path MTU Discovery (PMTUd) est une technique qui permet de découvrir la taille de MTU maximum qui permettra un transfert de paquets sans fragmentation, entre deux hôtes
- Ce sera la plus petite taille de MTU rencontrée sur le chemin entre les deux hôtes
- La commande tracepath permet de découvrir cette taille

  Exemple : tracepath google.com

Complément: le MTU : latence de transmission

- Soit un débit moyen de ligne = 1.544.000 bits/sec (193.000 car/sec)
- Soit 10 sauts (nbr de routeurs à traverser) de l'émetteur vers le destinataire
- Soit 1 Mo de données à transférer (1 048 576 carac.)
- Calculons le temps total de transfert pour 1 Mo de données à transférer avec deux MTU différents; 1500 et 576

Mtu	Trame		Latence	Latence totale	Nb trames pour 1 Mo	Tps total
	Data (MSS)	Header: 20(tcp) + 20 (ip)				
1500	1460	40	7,772 ms	7,772 * 10 = 77,72 ms	1.048.576 / 1460 = 719	719 * 77,72 ms = 55,88 sec
576	536	40	2.984 ms	2.984 * 10 = 29,84 ms	1.048.576 / 536 = 1957	1957 * 29,84 ms = 58,40 sec

Complément: le MTU : latence de transmission

- La latence est donnée par la formule:

  latence = (MSS + header) * 8 / débit de la ligne en bits/sec

  8 parce que 8 bits/byte

- On constate que la latence est plus faible si le MTU est plus petit … mais il faut plus de trames … donc le temps total est plus grand car on doit envoyer plus d'entêtes de trame

- Il vaut donc mieux avoir un MTU plus grand

MTU	Petit	Grand
Tps transfert théorique	Plus élévé	Plus faible
Tps de retransmission d'un paquet	Plus faible	Plus élevé

route: table de routage: visualisation

- route ou route -n permet de visualiser la table de routage du PC

```
root@srvLX01:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.1.1     0.0.0.0         UG    0      0        0 enp0s3
10.0.20.0       0.0.0.0         255.255.255.0   U     0      0        0 enp0s8
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 enp0s3
root@srvLX01:~#
```

- Informations affichées:
  - Destination: indique l'adresse IP d'hôte/réseau de destination (0.0.0.0 pour la destination par défaut quand il n'y a pas de correspondance avec une autre ligne de la table de routage)
  - Gateway: indique la passerelle de l'hôte/réseau de destination (0.0.0.0 pour indiquer qu'il n'y a pas de passerelle pour ce réseau)

route: table de routage: visualisation

```
root@srvLX01:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.1.1     0.0.0.0         UG    0      0        0 enp0s3
10.0.20.0       0.0.0.0         255.255.255.0   U     0      0        0 enp0s8
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 enp0s3
root@srvLX01:~#
```

- Informations affichées:
  - Genmask: indique le masque de réseau du réseau de destination (0.0.0.0 pour la route par défaut et 255.255.255.255 pour un hôte)
  - Flags: indique le statut actuel de route
    - U: la route est activée (UP)
    - H: la destination est un hôte (Host)
    - G: la destination est une passerelle (gateway)

route: table de routage: visualisation

```
root@srvLX01:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.1.1     0.0.0.0         UG    0      0        0 enp0s3
10.0.20.0       0.0.0.0         255.255.255.0   U     0      0        0 enp0s8
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 enp0s3
root@srvLX01:~#
```

- Informations affichées:
  - Metric: distance (en sauts) jusqu'à la destination (n'est plus utilisé)
  - Ref: nombre de références associées à la route (n'est plus utilisé)
  - Use: compteur d'utilisation de la route
  - Iface: nom de l'interface réseau associée à la route

route: table de routage: visualisation

```
root@srvLX01:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.1.1     0.0.0.0         UG    0      0        0 enp0s3
10.0.20.0       0.0.0.0         255.255.255.0   U     0      0        0 enp0s8
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 enp0s3
root@srvLX01:~#
```

- Informations affichées:
  - Tout paquet à destination du réseau 192.168.1.0/24 est envoyé directement à l'adresse concernée en passant par la carte enp0s3
  - Tout paquet à destination du réseau 10.0.20.0/24 est envoyé directement à l'adresse concernée en passant par la carte enp0s8
  - Tout autre paquet ne correspondant pas à une des lignes précédentes est envoyé à l'adresse 192.168.1.1 (passerelle) via enp0s3, c'est la route par défaut

route: route par défaut

- Pour ajouter une route par défaut:

  route add default gw *adresse_passerelle*
  Exemple: route add default gw 192.168.13.1

- Pour enlever une route par défaut (et la remplacer éventuellement par une autre):

  route del default gw *adresse_passerelle*
  Exemple: route del default gw 192.168.13.1

route: ajout d'une route

- Pour ajouter une route vers un réseau:

  route add -net *adresse_réseau* netmask *masque* gw *passerelle*
  Exemple:
  route add -net 192.168.7.0 netmask 255.255.255.0
  gw 192.168.13.1

- Pour ajouter une route vers un hôte:

  route add -host *adresse_hôte* gw *passerelle*
  Exemple:
  route add -host 192.168.7.10 gw 192.168.13.1

  remarque: le paramètre -host est optionnel

route: ajout d'une route

- Si nécessaire, on peut également préciser l'interface de sortie vers le réseau de destination:

  route add -net *adresse_réseau* netmask *masque* gw *passerelle*
  dev *interface*
  Exemple:
  route add -net 192.168.7.0 netmask 255.255.255.0
  gw 192.168.13.1 dev enp0s3

- route add -host *adresse_hôte* gw *passerelle* dev *interface*
  Exemple:
  route add -host 192.168.7.10 gw 192.168.13.1 dev enp0s3

route: ajout d'une route

- On peut aussi uniquement préciser l'interface de sortie vers le réseau de destination:

  route add -net *adresse_réseau* netmask *masque* dev *interface*
  Exemple:
  route add -net 192.168.7.0 netmask 255.255.255.0 dev enp0s3

- route add -host *adresse_hôte* dev *interface*
  Exemple:
  route add -host 192.168.7.10 dev enp0s3

route: suppression d'une route

- Pour supprimer une route vers un réseau:

  route del -net *adresse_réseau* netmask *masque* gw *passerelle*
  Exemple:
  route del -net 192.168.7.0 netmask 255.255.255.0
  gw 192.168.13.1

- Pour supprimer une route vers un hôte:

  route del -host *adresse_hôte* gw *passerelle*
  Exemple:
  route del -host 192.168.7.10 gw 192.168.13.1

  remarque: le paramètre -host est optionnel

route: atelier

- Ecrire les commandes permettant d'ajouter les routes suivantes:
  - pour toucher le réseau 192.168.8.0/24, les paquets doivent atteindre la passerelle 10.103.0.1 et sortir par enp0s3
  - pour toucher l'hôte 192.168.9.7, les paquets doivent atteindre la passerelle 10.103.0.1 et sortir par enp0s8
  - pour toucher le réseau 192.168.10.0/30, les paquets doivent sortir par enp0s3

route: atelier

- Ecrire les commandes permettant de supprimer toutes les routes ci-dessus ajoutées et celle par défaut
- Ecrire la commande permettant de rajouter la route par défaut
- La commande suivante aboutira-t-elle ? Oui ou non et pourquoi ?
  - # route add -net 192.168.23.0 netmask 255.255.255.0
    gw 192.168.15.1 enp0s3

route: atelier 2

- Comment supprimer de façon permanente la route du réseau "ZeroConf" (169.254.0.0/16) ?
- Vous trouverez plus d'info sur le "Zero Configuration Networking" sur: http://www.zeroconf.org
  http://fr.wikipedia.org/wiki/Zeroconf

route: remarque

- Les commandes précédentes ne sont valables que pendant une session et sont effacées au redémarrage
- Pour les conserver durablement, il faut les indiquer dans un fichier qui dans la plupart des distributions est:

  /etc/sysconfig/static-routes

  mais la syntaxe est un peu différente de celle vue précédemment …

netstat (**net**work **stat**istics)

- netstat est une commande permettant d'avoir des informations sur les connexions actives, les ports ouverts, les tables de routage, … et d'avoir des statistiques sur un certain nombre de protocoles
  - netstat -a permet de voir toutes les connexions TCP actives et les ports TCP et UDP sur lesquels l'ordinateur écoute
  - netstat -an idem mais sous forme numérique (les adresses IP ne sont pas résolues en nom de domaine)
  - netstat -r affiche la table de routage
  - netstat -rn idem mais sous forme numérique

netstat (**net**work **stat**istics)

- netstat -i donne les statistiques des interfaces réseau
- netstat -ie donne le même mais en mode étendu (équivalent à ifconfig -a)
- netstat -e donne les statistiques Ethernet
- netstat -s donne une liste de statistiques par protocole. Bon point de départ de débuggage si une application utilisateur devient lente …
- netstat -l donne la liste des sockets ouverts et des services qui sont à l'écoute)
- netstat -tu donne la liste des connexions TCP et UDP ouvertes
- netstat -tul donne la liste des services TCP et UDP qui sont à l'écoute
- netstat -tunl idem mais avec les numéros au lieu des noms
- netstat -tupan idem mais avec le nom des process correspondant au service
- …

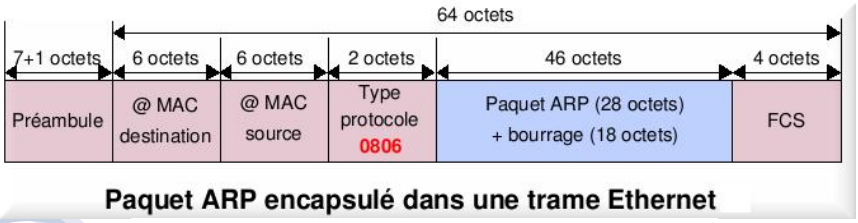
netstat (**net**work **stat**istics): atelier

- a) Quels sont les services TCP et UDP qui sont à l'écoute sur votre machine ?
- b) Quels sont les programmes liés à ces services ?
- c) Pourquoi le flag 'LISTEN' n'est-il pas présent pour les services UDP ?
- d) Connectez-vous sur le serveur 'ssh' du laboratoire.
- e) Affichez et interprétez la liste des connexions tcp.
- f) Quelle est la commande à entrer sur le serveur pour voir apparaître et disparaître en temps réel toutes les (dé)connexions ssh ?
- g) Réalisons l'expérience et observons l'état 'STATE' lors de chaque (dé)connexion... (man netstat)

arp

- Cette commande permet de voir et manipuler le cache ARP qui contient une table distincte par carte réseau
- Une entrée est systématiquement ajoutée par ARP lors de la réception de datagrammes si elle n'existait pas auparavant
- Les entrées dans le cache ont une courte durée de vie ...
- La commande ping met à jour le cache ARP

arp: trame



Paquet ARP encapsulé dans une trame Ethernet

ARP

Bit Number

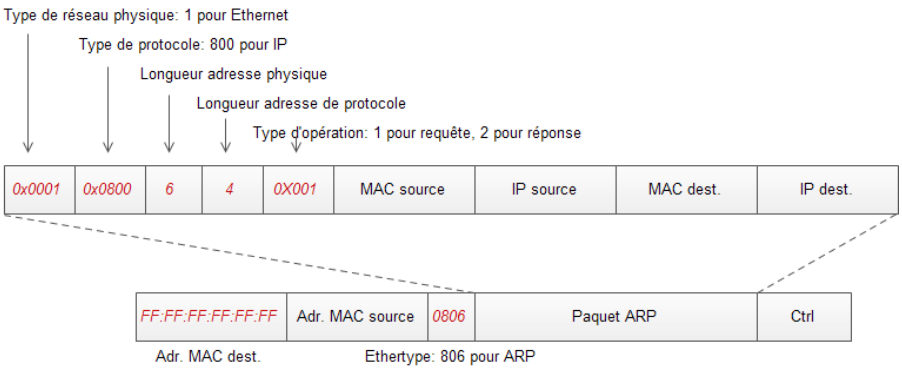
1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 3 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

Hardware Address Type		Protocol Address Type
H/w Addr Len	Prot. Addr Len	Operation
Source Hardware Address		
Source Hardware Addr (cont.)		Source Protocol Address
Source Protocol Addr (cont.)		Target Hardware Address
Target Hardware Address (cont.)		
Target Protocol Address		

Adresses MAC

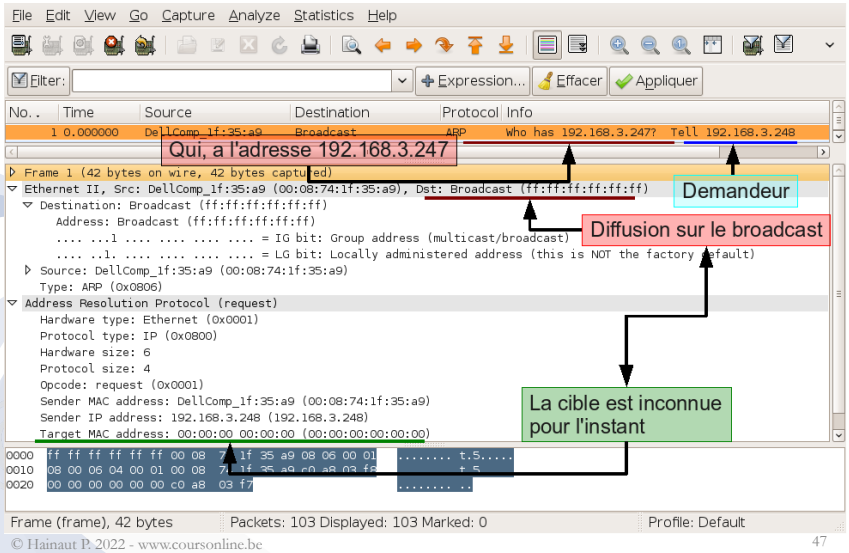
Correspondance entre IP et MAC

- Trame Ethernet contenant une requête ARP



Adresses MAC

Correspondance entre IP et MAC



arp: exemples

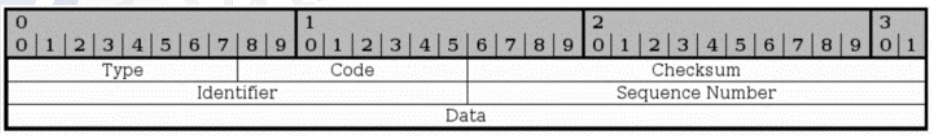
- arp -a affiche toutes les entrées du cache
- arp -a adresse_ip affiche uniquement l'entrée d'adresse_ip
- arp -s adresse_ip adresse_mac permet l'ajout manuel d'une entrée statique dans le cache (qui n'expire donc pas)
- arp -d adresse_ip permet la suppression d'une entrée statique

arp: atelier

- a) Demandez l'adresse IP de votre voisin
- b) Listez le contenu de votre cache ARP
- c) Pingez l'interface de votre voisin
- d) Listez le contenu de votre cache ARP
- e) Y a-t-il une différence ? Pourquoi ?
- f) Quelle différence fondamentale y a-t-il entre l'ARP et le DHCP ?

ping

- Terme emprunté aux sonars …
- Envoi d'un paquet de données et attente du retour du paquet (echo request / echo reply)
- Fonctionne si les 2 couches inférieures de la pile TCP/IP (Accès Réseau / Internet) sont opérationnelles
- Echo Request ou Echo Reply Header Format (RFC 792):



ping

Par défaut, des paquets 'echo request' / 'echo reply' de 64 bytes (56 de données + 8 d'en-tête icmp) sont envoyés

@IP ou nom DNS de la cible

Temps aller/retour du paquet ICMP:
- micro sec sur un LAN
- milli sec sur un WAN

```
root@server:/var/www/midi# ping 173.194.65.73
PING 173.194.65.73 (173.194.65.73) 56(84) bytes of data.
64 bytes from 173.194.65.73: icmp_req=1 ttl=47 time=30.5 ms
64 bytes from 173.194.65.73: icmp_req=2 ttl=47 time=30.1 ms
64 bytes from 173.194.65.73: icmp_req=3 ttl=47 time=29.4 ms
64 bytes from 173.194.65.73: icmp_req=4 ttl=47 time=29.1 ms
^C
--- 173.194.65.73 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3007ms
rtt min/avg/max/mdev = 29.152/29.820/30.522/0.540 ms
root@server:/var/www/midi#
```

Statistiques du ping: nb paquets transmis et reçus, pourcentage de perte, temps minimum et maximum, temps moyen ...

N° de séquence du paquet

Time To Live

ping: TTL (Time To Live)

- Nombre de routeurs maximum que le paquet peut traverser avant d'être éliminé
- Chaque fois qu'un paquet traverse un router, son TTL est décrémenté d'une unité
- Evite aux paquets égarés de boucler indéfiniment sur le réseau
- Le champ TTL est enregistré dans la couche Internet

ping: TTL (Time To Live)

- Exemple: Un paquet revient avec un TTL de 121

  Son TTL initial=128 (puissance de 2 directement supérieure à 121)

  => le paquet a traversé 7 routeurs (128-121) avant de toucher sa cible

ping: exercice

- a) Lancez un renifleur (tshark/wireshark) à l'écoute de paquets icmp sur votre machine
- b) Pingez l'interface de la machine de votre voisin
- c) A l'aide du renifleur:
  - Localisez un couple de trames 'echo reply' et 'echo request'
  - Analysez leur contenu

ping: atelier

- a) Combien de routeurs séparent votre machine de celle qui est héberge le site Web de la Haute Ecole ?
- b) Quelles sont les adresses IP de ces différents routeurs ?

ping: les messages

- **unknown host**

  la conversion des noms en adresses ne fonctionne pas correctement
  => voir système de résolution de noms
  (ex. # ping www.isattt.be)

- **network is unreachable**

  le réseau de destination n'est pas joignable
  (le paquet ne sait pas quelle route prendre)
  => voir table de routage
  (ex. On 'pingue' une adresse extérieure sans route par défaut)

ping: les messages

- **destination host unreachable**

  la route existe mais la machine ne répond pas
  => elle est mal configurée ou problème de connexion
  (ex. On 'pingue' une adresse du LAN qui n'existe pas)

ping: autres flags

- -s définit la taille en bytes du paquet à envoyer
- -c définit le nombre de paquets à envoyer
- Comparez le temps moyen aller/retour de 10 ping entre votre machine et celle d'adresse 212.68.208.163:
  - avec des paquets icmp de 64 bytes
  - avec des paquets icmp de 512 bytes

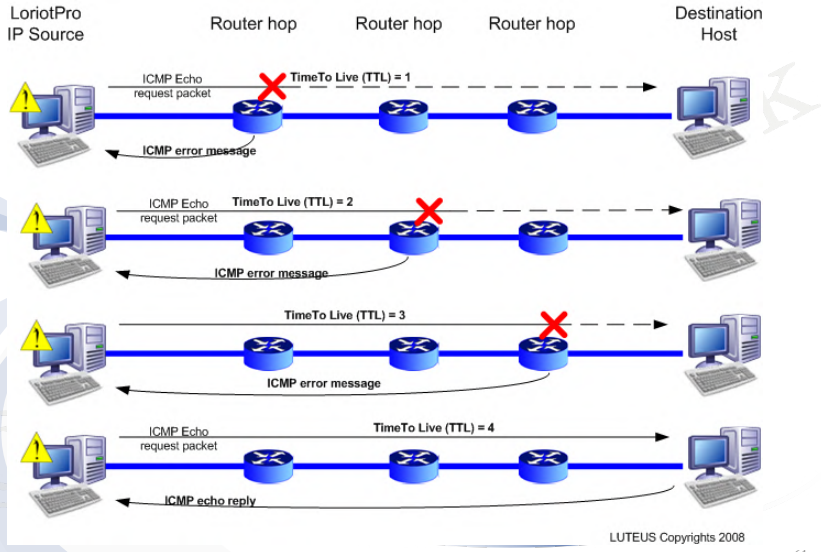
ping: autres flags

- -w délai durant lequel sont envoyés les paquets icmp (en secondes)
- -t définit le TTL de départ
- # ping -t 1 www.google.be -> Que se passe-t-il et pourquoi ?
  # ping -t 2 www.google.be -> Que se passe-t-il et pourquoi ?
  # ping -t 30 www.google.be -> Que se passe-t-il et pourquoi ?

traceroute

- Pour suivre le cheminement des paquets sur le réseau
- Si votre machine est configurée pour travailler avec un DNS, les réponses feront apparaître des noms de villes, de régions ou de messageries
- Principe de fonctionnement: ex. traceroute www.google.be

traceroute



Quelques fichiers particuliers

- **/proc/net/route** -> table de routage
- **/proc/net/arp** -> cache arp
- **/proc/sys/net/ipv4/ip_forward** -> (dés)active l'ip forwarding --> (0 ou) 1 (configuration d'une passerelle)
- **/proc/sys/net/ipv4/icmp_echo_ignore_all** -> (dés)active l'acceptation d'un ping --> (0 ou) 1
- **/proc/sys/net/ipv4/icmp_echo_ignore_broadcasts** -> (dés)active l'acceptation d'une requête diffusée icmp --> (0 ou) 1
- Tous ces fichiers sont stockés en Ram puisque dans /proc

Quelques fichiers particuliers

- Configuration d'un hôte en passerelle au bootage

  Le fichier **/etc/sysctl.conf** -> appelé par le script de démarrage du réseau **/etc/init.d/network**

  ...

  net.ipv4.ip_forward=x (avec x=0 ou 1)

- Atelier: Configurez votre machine en passerelle:
  - a) en dynamique
  - b) en statique
- Voir man sysctl

Quelques fichiers particuliers

- Fichier **/etc/hosts**

  127.0.0.1 Acer localhost.localdomain localhost
  198.197.56.9 magateway

- Fichier **/etc/networks**

  net1 198.197.56.0
  coursunix 191.12.1.128 alias1 #comment1

- Cela permet d'écrire

  route add -net coursunix netmask 255.255.255.128 gw mygateway enp0s3

Quelques fichiers particuliers

- Les fichiers

  /etc/services: table d'association numéros/noms de service

  /etc/protocols: table d'association numéros/noms de protocoles

  - Voir sur la machine
  - A ne pas modifier !! Sauf cas exceptionnel. Pourquoi ?

Fichiers de configuration

- Paramètres généraux: /etc/sysconfig/network

```
NETWORKING=yes
HOSTNAME=PcDemo
GATEWAY=193.190.156.1
GATEWAYDEV=enp0s3
```

C'est la route par défaut

- Paramètres spécifiques aux interfaces
  Fichiers /etc/sysconfig/network-scripts/ifcfg-*
  - /etc/sysconfig/network-scripts/ifcfg-enp0s3

```
DEVICE=enp0s3
ONBOOT=yes
BOOTPROTO=dhcp
```

IP dynamique

  - /etc/sysconfig/network-scripts/ifcfg-lo

```
DEVICE=lo
BOOTPROTO=static
IPADDR=127.0.0.1
NETMASK=255.0.0.0
NETWORK=127.0.0.0
BROADCAST=127.255.255.255
ONBOOT=yes
```

IP statique

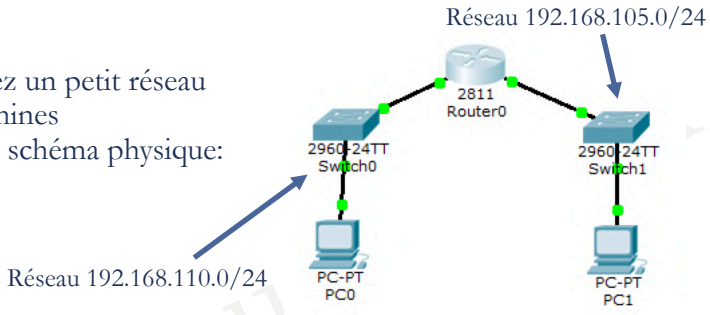
Paramètres facultatifs, ils seront calculés automatiquement

Fichiers de configuration: atelier

- Configuration au boot:
  - Quel est le nom du script de la configuration réseau déclenché lors du bootage ?
  - Comment démarrer / redémarrer / arrêter l'accès réseau ?
  - Après analyse du contenu de ce fichier, trouvez le nom du fichier qui nous permettra d'ajouter des routes statiques lors du bootage ?
  - Comment configurer ce fichier ?

Exercice 1

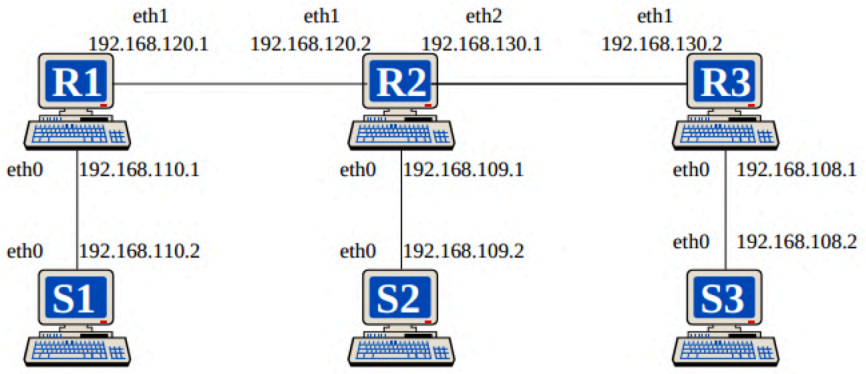
- Constituez un petit réseau de 3 machines
  Voici son schéma physique:



Router0 joue le rôle de passerelle entre les 2 autres et sera configurée dynamiquement
Pour PC0, la table de routage se construira automatiquement au démarrage
Pour PC1, la table de routage sera configurée dynamiquement
Router0, PC0 et PC1 sont des machines Linux

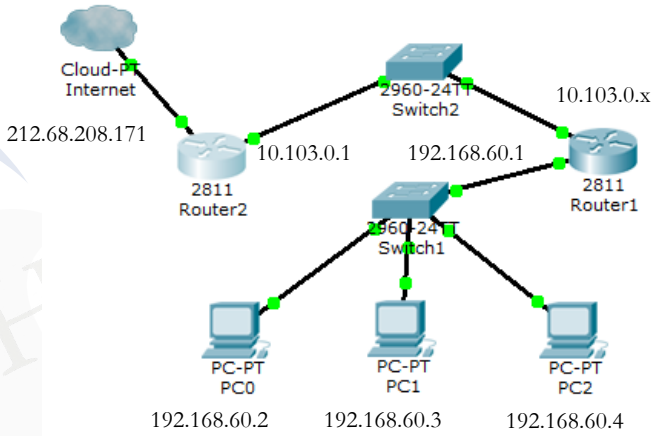
Exercice 2

- Configurez et testez la maquette suivante (6 machines):



Exercice 3

- Soit la maquette suivante:



Exercice 3: suite

- En sachant que Router2 fait office de routeur d'accès Internet pour le laboratoire:

  a) Quelles sont la (ou les) fonctionnalité(s) à activer sur Router1 ?

  b) Comment les activer ?

  c) Configurez tout le réseau (uniquement PC0 et Router1) en aval de Router2 pour que chaque machine puisse accéder à l'Internet

iproute

- Les commandes ifconfig et route n'exploitent toutes les possibilités offertes par les noyaux 2.4 et supérieurs
- Le paquetage iproute (ou iproute2 dans certaines distributions) permettra d'exploiter au mieux les fonctions de routage avancées de ces noyaux
- Ce paquetage contient plusieurs utilitaires
- Nous allons en aborder certains dans les dias suivantes

Equivalences: ifconfig

- ifconfig devient ip addr list ou ip a l
- ifconfig –a devient ip link list ou ip l l
- ifconfig enp0s3 devient ip addr show dev enp0s3 ou ip a s enp0s3

```
root@srvLX01:~# ip addr show enp0s3
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 08:00:27:ab:c4:43 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.123/24 brd 192.168.1.255 scope global enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:feab:c443/64 scope link
       valid_lft forever preferred_lft forever
root@srvLX01:~# _
```

- Informations affichées:
  - Essentiellement les mêmes que pour ifconfig

Equivalences: ifconfig

- Un -4 comme argument affiche uniquement les infos IPv4
  Exemple: ip -4 addr show enp0s3
- Un -6 comme argument affiche uniquement les infos IPv6
  Exemple: ip -6 addr show
- ip addr add *ip_address/netmask* dev *interface* assigne des paramètres IP (n'est effectif que durant la session, disparait au redémarrage)
  comme ifconfig *interface ip_address netmask*

  Exemple: ip addr add 10.0.20.1/255.255.255.0 dev enp0s8
  ou ip addr add 10.0.20.1/24 dev enp0s8

Equivalences: ifconfig

- Contrairement à ifconfig, on peut assigner plusieurs adresses IP à une interface
- ip addr del *ip_address/netmask* dev *interface* enlève des paramètres IP

  Exemple: ip addr del 10.0.20.1/255.255.255.0 dev enp0s8
  ou ip addr del 10.0.20.1/24 dev enp0s8

- Ip addr flush dev *interface* (ip a f *interface*) enlève tous les paramètres IP de l'interface
  Exemple: ip -4 a f enp0s8 enlèvera tous les paramètres IPv4

Equivalences: ifconfig

- ifconfig *interface* up devient ip link set dev *interface* up
  et permet d'activer une interface

  Exemple: ip l s enp0s8 up

- ifconfig *interface* down devient ip link set dev *interface* down
  et permet de désactiver une interface

  Exemple: ip l s enp0s8 down

Equivalences: ifconfig

- Pour trouver le MTU, ifconfig *interface* | grep -Eoi "MTU:[0-9]+" devient ip link show *interface* ou ip addr show *interface*
- Pour changer le MTU, ifconfig *interface* mtu *valeur_mtu* devient ip link set mtu *valeur_mtu* dev *interface*

  Exemple: ip l s mtu 1400 enp0s3

Equivalences: route

- route –n devient ip route (ip r) ou ip route show (ip r s)

  Les informations sont présentées différemment

```
root@srvLX01:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.1.1     0.0.0.0         UG    0      0        0 enp0s3
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 enp0s3
root@srvLX01:~#
root@srvLX01:~# ip r s
default via 192.168.1.1 dev enp0s3
192.168.1.0/24 dev enp0s3  proto kernel  scope link  src 192.168.1.123
root@srvLX01:~# _
```

Equivalences: route

- route add default gw *adresse_passerelle* devient
  ip route add default via *adresse_passerelle*

  Exemple: ip route add default via 192.168.13.1

- route del default gw *adresse_passerelle* devient
  ip route del default

Equivalences: route

- route add -net *adresse_réseau* netmask *masque* gw *passerelle* devient
  ip route add *adresse_réseau*/*masque* via *passerelle*
  Exemple:
  ip route add 192.168.7.0/255.255.255.0 via 192.168.13.1 ou
  ip route add 192.168.7.0/24 via 192.168.13.1

- route del -net *adresse_réseau* netmask *masque* gw *passerelle*
  devient
  ip route del *adresse_réseau*/*masque* via *passerelle*
  Exemple:
  ip route del 192.168.7.0/255.255.255.0 via 192.168.13.1 ou
  ip route del 192.168.7.0/24 via 192.168.13.1

Equivalences: arp

- arp –a devient ip neight show (ip n s)

```
root@srvLX01:~# ip n s
192.168.1.1 dev enp0s3 lladdr 00:1d:7e:31:f8:de STALE
root@srvLX01:~# _
```

```
root@srvLX01:~# ip n s
192.168.1.1 dev enp0s3 lladdr 00:1d:7e:31:f8:de REACHABLE
root@srvLX01:~# _
```

3 états possibles pour une entrée:
- reachable: valide et accessible
- stale: valide mais inaccessible (unreachable)
- delay: un paquet a été envoyé au voisin et le noyau attend la réponse

Equivalences: arp

- arp -s *adresse_ip adresse_mac* devient
  ip neight add *adr_ip* lladdr *adr_mac* dev *interface* nud *état*

où l'état peut prendre:
- *permanent* : entrée toujours valide
- *noapr* : entrée valide mais non validée, peut être retirée quand sa durée de vie expire
- *stale* : entrée valide mais suspecte
- *reachable* : entrée valide jusqu'à expiration du délai d'accessibilité

- Exemple: ip neigh add 10.0.20.5 lladdr 00:1e:fa:45:c3:00 dev enp0s8 nud perm

Equivalences: arp

- arp -d *adresse_ip* devient ip neight del *adr_ip* dev *interface*
  Exemple: ip neigh del 10.0.20.5 dev enp0s8
- ip neight chg *adr_ip* dev *interface* nud *état*
  permet de changer l'état de l'entrée
  Exemple: ip neigh chg 10.0.20.5 dev enp0s8 nud reacheable
- Ip –s –s neight flush *adresse_ip*
  permet de vider la table ARP d'une adresse IP
  Exemple: ip –s –s n f 10.0.20.5

Conclusion

- Les commandes de base réseau n'ont plus de secret pour vous
- Vous voilà donc paré pour configurer des serveurs sous Linux
- Merci de votre attention