

## TP 2

### Administration réseau sur Linux

*Viet NGUYEN -- 20006303*

#### A. La command ping:

Après lancer la commande **\$ ping** voici mon résultat:

```
viet@vietLaptop:~$ ping univ-paris8.fr
PING univ-paris8.fr (193.54.155.1) 56(84) bytes of data.
64 bytes from up8.univ-paris8.fr (193.54.155.1): icmp_seq=1 ttl=50 time=17.0 ms
64 bytes from up8.univ-paris8.fr (193.54.155.1): icmp_seq=2 ttl=50 time=19.4 ms
64 bytes from ns.univ-paris8.fr (193.54.155.1): icmp_seq=4 ttl=50 time=15.8 ms
64 bytes from ns.univ-paris8.fr (193.54.155.1): icmp_seq=5 ttl=50 time=12.0 ms
64 bytes from up8.univ-paris8.fr (193.54.155.1): icmp_seq=6 ttl=50 time=14.3 ms
64 bytes from ns.univ-paris8.fr (193.54.155.1): icmp_seq=7 ttl=50 time=14.2 ms
64 bytes from ns.univ-paris8.fr (193.54.155.1): icmp_seq=8 ttl=50 time=20.8 ms
64 bytes from ns.univ-paris8.fr (193.54.155.1): icmp_seq=9 ttl=50 time=13.3 ms
64 bytes from up8.univ-paris8.fr (193.54.155.1): icmp_seq=10 ttl=50 time=17.6 ms
^C
--- univ-paris8.fr ping statistics ---
10 packets transmitted, 9 received, 10% packet loss, time 9017ms
rtt min/avg/max/mdev = 11.955/16.021/20.762/2.741 ms
viet@vietLaptop:~$
```

1. L'adresse IP du serveur [univ-paris8.fr](http://univ-paris8.fr) est **193.54.155.1** .
2. TTL signifie "Time To Live" et représente le nombre de sauts (routers) qu'un paquet peut prendre avant d'être abandonné. Il est utilisé pour éviter que les paquets ne tournent en boucle indéfiniment en cas de routage incorrect.  
"Time" représente le temps de réponse du paquet en millisecondes (ms), c'est-à-dire le temps nécessaire pour que le paquet atteigne le serveur et que la réponse revienne.
3. Le taux de perte des paquets n'est pas indiqué dans les résultats fournis. La commande ping fournit le temps de réponse pour chaque paquet envoyé au serveur, mais elle ne fournit pas le nombre de paquets envoyés ou reçus. Pour

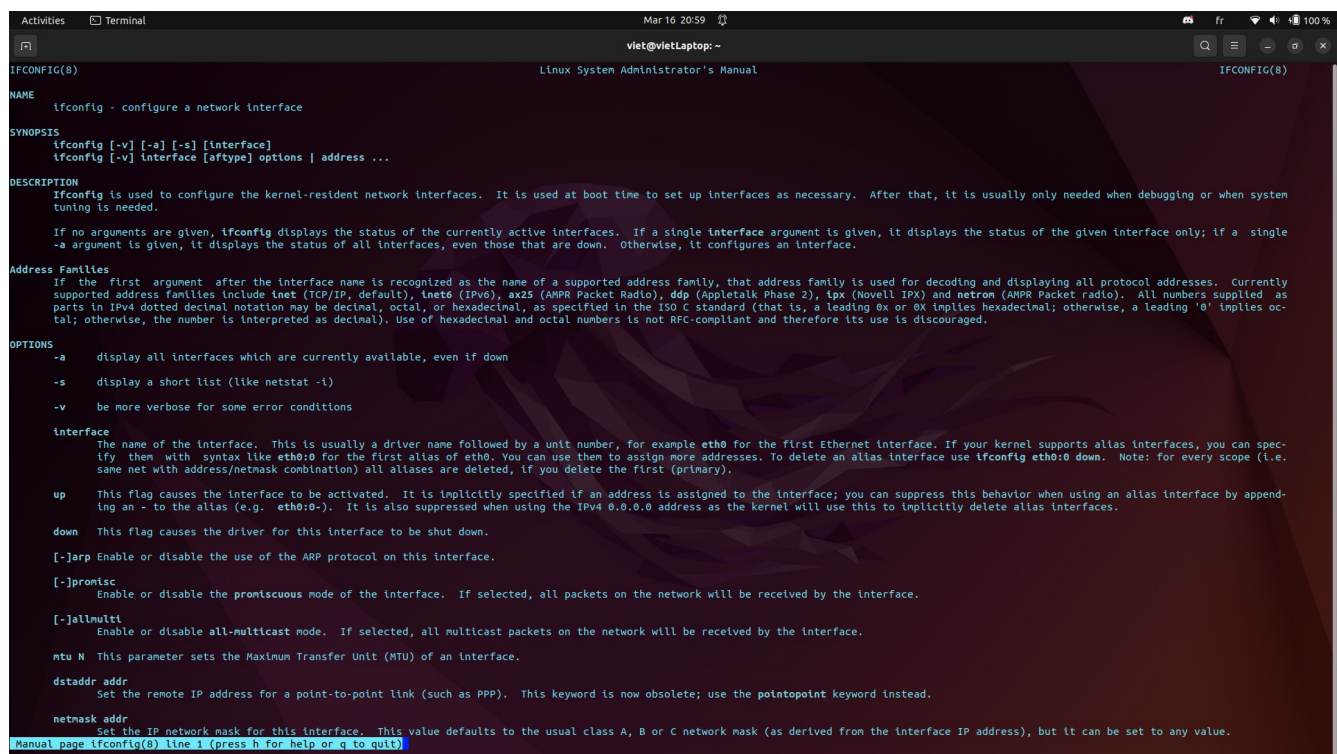
calculer le taux de perte des paquets, il faut connaître le nombre de paquets envoyés et le nombre de paquets reçus qui sont indiqués à la fin de l'exécution de la commande.

Par exemple: "5 packets transmitted, 5 received, 0% packet loss". Si le taux de perte des paquets est de 0%, cela signifie que tous les paquets ont été envoyés et reçus sans perte.

## B. La command ifconfig:

1)

Voici le resultat de la commande **\$ man ifconfig** , la documentation complète de la commande **ifconfig** :



```
Activities Terminal Mar 16 20:59 viet@vietLaptop: ~
ifconfig(8) Linux System Administrator's Manual ifconfig(8)

NAME
    ifconfig - configure a network interface

SYNOPSIS
    ifconfig [-v] [-a] [-s] [interface]
    ifconfig [-v] interface [aftype] options | address ...

DESCRIPTION
    Ifconfig is used to configure the kernel-resident network interfaces. It is used at boot time to set up interfaces as necessary. After that, it is usually only needed when debugging or when system tuning is needed.

    If no arguments are given, ifconfig displays the status of the currently active interfaces. If a single interface argument is given, it displays the status of the given interface only; if a single -a argument is given, it displays the status of all interfaces, even those that are down. Otherwise, it configures an interface.

Address Families
    If the first argument after the interface name is recognized as the name of a supported address family, that address family is used for decoding and displaying all protocol addresses. Currently supported address families include inet (TCP/IP, default), inet6 (IPv6), ax25 (AMPR Packet Radio), ddp (Appletalk Phase 2), ipx (Novell IPX) and netrom (AMPR Packet radio). All numbers supplied as parts in IPv4 dotted decimal notation may be decimal, octal, or hexadecimal, as specified in the ISO C standard (that is, a leading 0x or 0X implies hexadecimal; otherwise, a leading '0' implies octal; otherwise, the number is interpreted as decimal). Use of hexadecimal and octal numbers is not RFC-compliant and therefore its use is discouraged.

OPTIONS
    -a    display all interfaces which are currently available, even if down
    -s    display a short list (like netstat -i)
    -v    be more verbose for some error conditions

Interface
    The name of the interface. This is usually a driver name followed by a unit number, for example eth0 for the first Ethernet interface. If your kernel supports alias interfaces, you can specify them with syntax like eth0:0 for the first alias of eth0. You can use them to assign more addresses. To delete an alias interface use ifconfig eth0:0 down. Note: for every scope (i.e. same net with address/netmask combination) all aliases are deleted, if you delete the first (primary).

up
    This flag causes the interface to be activated. It is implicitly specified if an address is assigned to the interface; you can suppress this behavior when using an alias interface by appending an - to the alias (e.g. eth0:0-). It is also suppressed when using the IPv4 0.0.0.0 address as the kernel will use this to implicitly delete alias interfaces.

down
    This flag causes the driver for this interface to be shut down.

[-]arp
    Enable or disable the use of the ARP protocol on this interface.

[-]promisc
    Enable or disable the promiscuous mode of the interface. If selected, all packets on the network will be received by the interface.

[-]allmulti
    Enable or disable all-multicast mode. If selected, all multicast packets on the network will be received by the interface.

mtu N
    This parameter sets the Maximum Transfer Unit (MTU) of an interface.

dstaddr addr
    Set the remote IP address for a point-to-point link (such as PPP). This keyword is now obsolete; use the pointopoint keyword instead.

netmask addr
    Set the IP network mask for this interface. This value defaults to the usual class A, B or C network mask (as derived from the interface IP address), but it can be set to any value.

Manual page ifconfig(8) line 1 (press h for help or q to quit)
```

cliquez 'q' pour quitter.

2)

Pour afficher les informations de configuration de toutes les adresses IP et interfaces réseau, on peut utiliser la commande suivante:

**\$ ifconfig -a**

```

viet@vietLaptop:~$ ifconfig -a
enp0s31f6: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether 8c:ec:4b:e5:08:c8 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 16 memory 0xef200000-ef220000

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 111360 bytes 10621725 (10.6 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 111360 bytes 10621725 (10.6 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlp2s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.10 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::53d2:1370:9174:3db8 prefixlen 64 scopeid 0x20<link>
    ether e4:70:b8:7d:0b:f9 txqueuelen 1000 (Ethernet)
    RX packets 251045 bytes 332577976 (332.5 MB)
    RX errors 0 dropped 4 overruns 0 frame 0
    TX packets 88937 bytes 12973049 (12.9 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

viet@vietLaptop:~$

```

3) Pour récupérer l'adresse IP de l'interface réseau qui permet à notre poste de communiquer, on peut utiliser la commande suivante :

**\$ ifconfig [nom-interface]**

Remplacez [nom-interface] par le nom de notre interface réseau. L'adresse IP sera affichée à côté de "inet".

4) Le masque du réseau peut être trouvé en utilisant la même commande que pour récupérer l'adresse IP de l'interface réseau:

**\$ ifconfig [nom-interface]**

Le masque du réseau sera affiché à côté de "netmask".

5) L'adresse du réseau peut être calculée en appliquant le masque de réseau à l'adresse IP.

Par exemple, si l'adresse IP est 192.168.0.0.10 et le masque de réseau est 255.255.255.0, l'adresse du réseau est 192.168.0.255

6) Le nombre de bits réservés pour l'adressage de la partie hôte de notre adresse peut être calculé en utilisant la formule  $2^n - 2$ , où  $n$  est le nombre de bits dans la partie hôte.

Par exemple, si l'adresse IP est 192.168.0.10 et le masque de réseau est 255.255.255.0, alors 16 bits sont réservés pour l'adressage de la partie hôte (car le masque de réseau a 16 bits à 1), donc le nombre de bits pour la partie hôte est  $32 - 16 = 16$ .

Ainsi, le nombre d'adresse disponibles pour l'adressage de la partie hôte est  $2^{16} - 2 = 65,534$ .

7) Pour configurer l'adresse IP de notre machine avec la commande **ifconfig**, on peut utiliser la commande suivante :

**\$ ifconfig [nom-interface] [adresse-ip] netmask [masque-reseau]**

Remplacez [nom-interface] par le nom de notre interface réseau,

[adresse-ip] par l'adresse IP souhaitée,

[masque-reseau] par le masque de réseau souhaité.

Par exemple, pour attribuer l'adresse IP 192.168.0.10 et un masque de réseau 255.255.255.0 à l'interface réseau "eth0", la commande serait:

**\$ ifconfig eth0 192.168.0.10 netmask 255.255.255.0**

8) Pour changer la valeur MTU de notre interface réseau, utilisez la commande suivante:

**\$ ifconfig [nom-interface] mtu [valeur-mtu]**

Remplacez [nom-interface] par le nom de notre interface réseau,

[valeur-mtu] par la valeur MTU souhaitée.

Par exemple, pour fixer la valeur MTU de l'interface "eth0" à 1500, la commande serait :

**\$ ifconfig eth0 mtu 1500**



## C. La command nslookup:

La commande : **\$ nslookup google.com**

```
viet@vietLaptop:~$ nslookup google.com
Server:          127.0.0.53
Address:         127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 216.58.213.78
Name:   google.com
Address: 2a00:1450:4007:80e::200e

viet@vietLaptop:~$
```

1)

L'adresse IP du serveur [google.com](https://www.google.com) est **216.58.213.78**.

2)

Il n'est pas possible de déterminer le nom de domaine de l'hôte ayant l'adresse IP **193.54.174.1** à partir de la sortie de la commande **\$ nslookup google.com**.

Pour trouver le nom de domaine d'une adresse IP donnée, on peut utiliser la commande inverse **nslookup** en spécifiant l'adresse IP comme argument.

Par exemple, pour trouver le nom de domaine de l'adresse IP **193.54.174.1**, on peut exécuter la commande :

```
viet@vietLaptop:~$ nslookup 193.54.174.1
1.174.54.193.in-addr.arpa      name = ns2.univ-paris8.fr.

Authoritative answers can be found from:

viet@vietLaptop:~$
```

Dans ce cas, le nom de domain de l'hote est **ns2.univ-paris8.fr**.

## D. La command netstat:

## 1) La commande **\$ netstat -help** :

```
viet@vietLaptop:~$ netstat -help
usage: netstat [-vWeenNcCF] [<Af>] -r          netstat {-V|--version|-h|--help}
       netstat [-vWnNcaeol] [<Socket> ...]
       netstat { [-vWeenNac] -i | [-cnNe] -M | -s [-6tuw] }

    -r, --route           display routing table
    -i, --interfaces      display interface table
    -g, --groups          display multicast group memberships
    -s, --statistics      display networking statistics (like SNMP)
    -M, --masquerade      display masqueraded connections

    -v, --verbose         be verbose
    -W, --wide            don't truncate IP addresses
    -n, --numeric         don't resolve names
    --numeric-hosts       don't resolve host names
    --numeric-ports       don't resolve port names
    --numeric-users       don't resolve user names
    -N, --symbolic        resolve hardware names
    -e, --extend          display other/more information
    -p, --programs        display PID/Program name for sockets
    -o, --timers          display timers
    -c, --continuous      continuous listing

    -l, --listening       display listening server sockets
    -a, --all             display all sockets (default: connected)
    -F, --fib             display Forwarding Information Base (default)
    -C, --cache           display routing cache instead of FIB
    -Z, --context         display SELinux security context for sockets

<Socket>={-t|--tcp} {-u|--udp} {-U|--udplite} {-S|--sctp} {-w|--raw}
          {-x|--unix} --ax25 --ipx --netrom
<AF>=Use '-6|-4' or '-A <af>' or '--<af>'; default: inet
List of possible address families (which support routing):
  inet (DARPA Internet) inet6 (IPv6) ax25 (AMPR AX.25)
  netrom (AMPR NET/ROM) ipx (Novell IPX) ddp (Appletalk DDP)
  x25 (CCITT X.25)
viet@vietLaptop:~$
```

## 2)

Pour afficher la table de routage, on peut utiliser l'option **-r** ou **--route** avec la commande **netstat**. La syntaxe de base est la suivante :

### **\$ netstat -r**

Cela affichera la table de routage complète sur notre système. On peut également utiliser l'option **-n** ou **--numeric** pour afficher les adresses IP sous forme numérique plutôt que sous forme de noms de domaine.

### **\$ netstat -nr**

Cela affichera la table de routage en affichant les adresses IP numériques plutôt que les noms de domaine.

Il convient de noter que selon le système d'exploitation utilisé, la commande **netstat** peut être remplacée ou obsolète, il est donc recommandé de vérifier la documentation spécifique à notre système pour plus d'informations.

3)

Pour afficher les statistiques réseau, on peut utiliser la commande **netstat** avec l'option **-s** ou **--statistics**. Pour obtenir le nombre de connexions TCP actives, on peut utiliser la commande suivante :

**\$ netstat -s**

Pour obtenir le nombre de paquets UDP reçus et envoyés, nous pouvons utiliser la commande suivante :

**\$ netstat -su**

Cela affichera les statistiques de l'interface UDP sous la forme suivante :

```
viet@vietLaptop:~$ netstat -su
IcmpMsg:
  InType0: 4
  InType3: 5
  OutType3: 7
  OutType8: 8
Udp:
  6139 packets received
  4 packets to unknown port received
  0 packet receive errors
  4657 packets sent
  0 receive buffer errors
  0 send buffer errors
  IgnoredMulti: 28
UdpLite:
IpExt:
  InMcastPkts: 701
  OutMcastPkts: 139
  InBcastPkts: 28
  OutBcastPkts: 1
  InOctets: 11070455
  OutOctets: 4549252
  InMcastOctets: 123710
  OutMcastOctets: 14640
  InBcastOctets: 4684
  OutBcastOctets: 78
  InNoECTPkts: 42364
MPTcpExt:
viet@vietLaptop:~$
```

Remplacez "6139" par le nombre de paquets UDP reçus et  
"4657" par le nombre de paquets UDP envoyés.

4)

Pour afficher toutes les connexions réseau, y compris les connexions TCP et UDP actives, on peut utiliser la commande **netstat** avec l'option **-a** ou **--all**.

La syntaxe est la suivante :

**\$ netstat -a**

Cela affichera toutes les connexions réseau actives sur notre système, y compris les connexions TCP et UDP, avec les adresses IP et les ports associés.

Nous pouvons également utiliser l'option **-n** ou **--numeric** pour afficher les adresses IP sous forme numérique plutôt que sous forme de noms de domaine.

La commande complète serait alors :

**\$ netstat -an**

5)

Voici le résultat avec la commande **\$ netstat -l** :

```
vl@vletLaptop:~$ netstat -l
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN
tcp        0      0 localhost:39811          0.0.0.0:*               LISTEN
tcp        0      0 localhost:6463           0.0.0.0:*               LISTEN
tcp        0      0 localhost:1123           0.0.0.0:*               LISTEN
tcp        0      0 localhost:ipp            0.0.0.0:*               LISTEN
tcp        0      0 localhost:49152          0.0.0.0:*               LISTEN
tcp        0      0 *:::localhost:ipp       [::]:*                  LISTEN
udp        0      0 0.0.0.0:dnsmasq         0.0.0.0:*               *
udp        0      0 localhost:domain        0.0.0.0:*               *
udp        0      0 0.0.0.0:49284           0.0.0.0:*               *
udp        0      0 0.0.0.0:631             0.0.0.0:*               *
udp6      0      0 [::]:dnsmasq            [::]:*                  *
udp6      0      0 [::]:55054              [::]:*                  *
raw6      0      0 [::]:lpmv6-icmp         [::]:*                  *
```

Et

```
Active UNIX domain sockets (only servers)
Proto Refcnt Flags Type State I-Node Path
unix 2 [ ACC ] STREAM LISTENING 42140 @/tmp/.ICE-unix/1158
unix 2 [ ACC ] STREAM LISTENING 25452 /run/irqbalance/irqbalance642.sock
unix 2 [ ACC ] STREAM LISTENING 30323 /tmp/.ICE-unix/1158
unix 2 [ ACC ] STREAM LISTENING 31103 @/tmp/.X11-unix/X0
unix 2 [ ACC ] STREAM LISTENING 31105 @/tmp/.X11-unix/X1
unix 2 [ ACC ] STREAM LISTENING 31104 /tmp/.X11-unix/X0
unix 2 [ ACC ] STREAM LISTENING 31106 /tmp/.X11-unix/X1
unix 2 [ ACC ] STREAM LISTENING 30870 /run/user/1000/symlink/private
unix 2 [ ACC ] STREAM LISTENING 30876 /run/user/1000/bus
unix 2 [ ACC ] STREAM LISTENING 29659 @/tmp/dbus-qch0btG6
unix 2 [ ACC ] STREAM LISTENING 30878 /run/user/1000/gnupg/S.dirmngr
unix 2 [ ACC ] STREAM LISTENING 30880 /run/user/1000/gnupg/S.gpg-agent.browser
unix 2 [ ACC ] STREAM LISTENING 30882 /run/user/1000/gnupg/S.gpg-agent.extra
unix 2 [ ACC ] STREAM LISTENING 30884 /run/user/1000/gnupg/S.gpg-agent.ssh
unix 2 [ ACC ] STREAM LISTENING 30886 /run/user/1000/gnupg/S.gpg-agent
unix 2 [ ACC ] STREAM LISTENING 30888 /run/user/1000/pipefire-0
unix 2 [ ACC ] STREAM LISTENING 30890 /run/user/1000/pk-debconf-socket
unix 2 [ ACC ] STREAM LISTENING 30892 /run/user/1000/pulse/native
unix 2 [ ACC ] STREAM LISTENING 30894 /run/user/1000/snapd-session-agent.socket
unix 2 [ ACC ] STREAM LISTENING 31928 /run/user/1000/keyring/control
unix 2 [ ACC ] STREAM LISTENING 30283 /run/user/1000/keyring/ssh
unix 2 [ ACC ] STREAM LISTENING 31000 /run/user/1000/keyring/pkcs11
unix 2 [ ACC ] STREAM LISTENING 32451 @/home/vlet/.cache/lbus/dbus-Q2IKxRc9
unix 2 [ ACC ] STREAM LISTENING 31033 /run/user/1000/at-spi/bus
unix 2 [ ACC ] STREAM LISTENING 31107 /run/user/1000/wayland-0
unix 2 [ ACC ] STREAM LISTENING 1181 /run/systemd/private
unix 2 [ ACC ] STREAM LISTENING 1183 /run/systemd/userdb/io.systemd.DynamicUser
unix 2 [ ACC ] STREAM LISTENING 1184 /run/systemd/io.systemd.ManagedOOM
unix 2 [ ACC ] STREAM LISTENING 1196 /run/systemd/fsck.progress
unix 2 [ ACC ] STREAM LISTENING 1287 /run/systemd/journal/stdout
unix 2 [ ACC ] STREAM LISTENING 1209 /run/udev/control
unix 2 [ ACC ] STREAM LISTENING 1240 /run/systemd/journal/io.systemd.journal
unix 2 [ ACC ] STREAM LISTENING 35555 /run/user/1000/vscode-52433f8c-1.76-main.sock
unix 2 [ ACC ] STREAM LISTENING 41610 /run/user/1000/vscode-glt-926896fb9.sock
unix 2 [ ACC ] STREAM LISTENING 20870 /run/systemd/resolve/io.systemd.Resolve
unix 2 [ ACC ] STREAM LISTENING 27083 /run/acpid.socket
unix 2 [ ACC ] STREAM LISTENING 27085 /run/avahi-daemon/socket
unix 2 [ ACC ] STREAM LISTENING 27087 /run/cups/cups.sock
unix 2 [ ACC ] STREAM LISTENING 27089 /run/dbus/system_bus_socket
unix 2 [ ACC ] STREAM LISTENING 27091 /run/snapd.socket
unix 2 [ ACC ] STREAM LISTENING 27093 /run/snapd-snap.socket
unix 2 [ ACC ] STREAM LISTENING 27095 /run/uuid/request
unix 2 [ ACC ] STREAM LISTENING 42358 /run/user/1000/snap.discord.org.chronlun.ChronLun.YyTY39/SS
unix 2 [ ACC ] STREAM LISTENING 40621 /run/user/1000/snap.discord/discord-ipc-0
unix 2 [ ACC ] STREAM LISTENING 29660 @/tmp/dbus-yL7V1SS
```



## E. La commande arp:

La commande **\$ sudo arp** :

```
viet@vietLaptop:~$ sudo arp
Address          HWtype  HWaddress      Flags Mask    Iface
_gateway         ether    ac:3b:77:51:d2:82  C             wlp2s0
viet@vietLaptop:~$
```

1)

La table ARP (Address Resolution Protocol) contient les adresses MAC et IP correspondantes pour les machines de notre réseau. Elle est utilisée pour résoudre les adresses IP des machines en adresses MAC pour permettre la communication sur le réseau.

2)

Pour afficher les adresses IP à la place des noms d'hôte et affiche un résumé avec le nombre d'entrée et ceux ignorés :

```
viet@vietLaptop:~$ sudo arp -n -v
Entries: 0      Skipped: 0      Found: 0
viet@vietLaptop:~$
```

3)

Le nombre d'entrées dans la table ARP dépend de notre réseau et du nombre de machines qui y sont connectées. Pour connaître le nombre d'entrées dans la table ARP de notre machine, on peut utiliser la commande suivante avec l'option **-a** :

**\$ sudo arp -a**

Cela affichera toutes les entrées de la table ARP avec les adresses IP et MAC correspondantes pour chaque machine de notre réseau. Le nombre total d'entrées dans la table ARP sera le nombre de lignes affichées par la commande.

## F. La commande route:

1) **\$ route -n**

```
viet@vietLaptop:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.0.1    0.0.0.0         UG    600    0      0 wlp2s0
169.254.0.0     0.0.0.0        255.255.0.0     U     1000   0      0 wlp2s0
192.168.0.0     0.0.0.0        255.255.255.0   U     600    0      0 wlp2s0
viet@vietLaptop:~$
```

2) En supprimant l'option **-n**, la commande **\$route** affiche les noms des réseaux plutôt que leurs adresses IP sous la colonne Destination.

```
viet@vietLaptop:~$ route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
default _gateway 0.0.0.0 UG 600 0 0 wlp2s0
link-local 0.0.0.0 255.255.0.0 U 1000 0 0 wlp2s0
192.168.0.0 0.0.0.0 255.255.255.0 U 600 0 0 wlp2s0
viet@vietLaptop:~$
```

## G. La commande traceroute:

La commande **traceroute** permet de tracer le chemin emprunté par les paquets IP depuis la machine locale jusqu'à une destination spécifiée. Elle affiche une liste ordonnée de tous les routeurs traversés, ainsi que les temps de réponse des paquets.

La syntaxe de base est la suivante :

**\$ traceroute [adresse\_IP ou nom\_de\_domaine]**

En voici un exemple :

```
viet@vietLaptop:~$ traceroute google.com
traceroute to google.com (142.250.178.142), 30 hops max, 60 byte packets
 1 _gateway (192.168.0.1) 6.656 ms 26.048 ms 25.969 ms
 2 10.16.0.1 (10.16.0.1) 26.778 ms 26.700 ms 26.626 ms
 3 chairj-ge-1-1-0.200.numericable.net (213.245.254.129) 26.551 ms 26.473 ms 26.396 ms
 4 177.117.223.213.rev.sfr.net (213.223.117.177) 26.894 ms 31.964 ms 30.797 ms
 5 71.146.6.194.rev.sfr.net (194.6.146.71) 31.812 ms 32.284 ms 35.400 ms
 6 71.146.6.194.rev.sfr.net (194.6.146.71) 34.746 ms 11.106 ms 10.093 ms
 7 74.125.146.198 (74.125.146.198) 17.204 ms 15.706 ms 20.382 ms
 8 108.170.244.193 (108.170.244.193) 18.759 ms * 18.530 ms
 9 108.170.244.225 (108.170.244.225) 19.259 ms 216.239.48.44 (216.239.48.44) 18.373 ms 142.251.64.124 (142.251.64.124) 22.622 ms
10 par21s22-in-f14.1e100.net (142.250.178.142) 22.546 ms 21.897 ms 27.223 ms
viet@vietLaptop:~$
```

Chaque ligne correspond à un routeur traversé sur le chemin vers la destination, ainsi qu'au temps de réponse des paquets. Les colonnes indiquent :

- le numéro de saut (hop) : de 1 à 30 par défaut
- l'adresse IP du routeur
- l'identifiant DNS du routeur (s'il existe)
- le temps de réponse des paquets (en millisecondes)

## H. La commande who:

1)

La commande **\$who** affiche la liste des utilisateurs actuellement connectés sur le système. Pour chaque utilisateur, elle affiche son nom d'utilisateur, le terminal qu'il utilise et l'heure à laquelle il s'est connecté.

2)

Pour déterminer combien d'utilisateurs sont connectés, il suffit de compter le nombre de lignes affichées par la commande **\$who**.

Par exemple, si la commande affiche une ligne, cela signifie qu'il y a un seul utilisateur connecté, comme ça:

```
viet@vietLaptop:~$ who
viet      tty2          2023-03-17 18:24 (tty2)
viet@vietLaptop:~$
```

## I. Les numéros de port:

Voici les numéros de port réservés pour les services demandés :

- World Wide Web HTTP : 80/tcp
- HTTP protocol over TLS/SSL : 443/tcp
- SSH Remote Login Protocol : 22/tcp
- Simple Mail Transfer (SMTP) : 25/tcp
- Echo : 7/tcp
- Daytime : 13/tcp
- Telnet : 23/tcp
- FTP (File Transfer Protocol) : 21/tcp