



# Network Management

## interface configuration

**DE HOGESCHOOL  
MET HET NETWERK**

Hogeschool PXL – Dep. PXL-IT – Elfde-Liniestraat 26 – B-3500 Hasselt  
[www.pxl.be](http://www.pxl.be) - [www.pxl.be/facebook](http://www.pxl.be/facebook)



# Intro

- configuratie nic:
  - via GUI: is mogelijk,  
kan problemen geven als simultaan in CLI wordt geconfigureerd
  - via CLI: varieert afhankelijk van distributie  
we bekijken Ubuntu in slides



# /etc/network/interfaces

```
student@server1:~$ cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp
```

2 interfaces gedefinieerd: lo en eth0

**auto**: auto-start (ifup -a)

**inet**: tcp/ip (inet6 = ipv6, ddp = apple, ipx = novell, ...)



# /etc/network/interfaces

```
student@server1:~$ cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp
```

lo = loopback device

virtueel device (geen hardware)

loopback: traffic naar dit device wordt gestuurd naar een service op OS

bvb. webserver localhost ip=127.0.0.1



# /etc/network/interfaces

```
student@server1:~$ cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp
```

eth0 = eerste netwerk (ethernet) kaart

optie 1: ip address via DHCP-server



# /etc/network/interfaces

```
student@server1:~$ cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.14.100
network 192.168.14.0
netmask 255.255.255.0
gateway 192.168.14.2
dns-nameservers 192.168.14.2
dns-search pxl.be
```

eth0 = eerste netwerk (ethernet) kaart

**optie 2:** fixed ip (static)

gateway: alle trafic voor buiten het netwerk wordt naar de gateway (router) gestuurd



# /etc/network/interfaces

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto ens33
iface ens33 inet dhcp
```

In de laatste versies van ubuntu zijn de netwerkkarten niet meer eth0, eth1, eth2, enz. genoemd



**ens33**: is de enige netwerkkart. Zou vroeger dus eth0 geweest zijn.

# /etc/network/interfaces

Vroeger werden de netwerkkarten genummerd in de volgorde dat de kernel deze zag tijdens het booten. Daarom kon het soms zijn dat eth0 en eth1 ineens gewisseld waren!

Nu wordt iedere netwerkkart tijdens het booten hernoemd naar een eenduidige naam, die steeds hetzelfde is.

en -- ethernet                      sl -- serial line IP (slip)                      wl -- wlan                      ww -- wwan

b<number> -- BCMA bus core number                      ccw<name> -- CCW bus group name

o<index>[d<dev\_port>] -- on-board device index number

s<slot>[f<function>][d<dev\_port>] -- hotplug slot index number

x<MAC> -- MAC address

```
$ dmesg | grep -i eth
```

```
[ 3.050064] e1000 0000:02:01.0 eth0: (PCI:66MHz:32-bit) 00:0c:29:05:a3:e2
[ 3.050074] e1000 0000:02:01.0 eth0: Intel(R) PRO/1000 Network Connection
[ 3.057410] e1000 0000:02:01.0 ens33: renamed from eth0
```





# /etc/network/interfaces

Het hernoemen van de netwerkkarten naar een naam die steeds hetzelfde blijft is een goed idee. Indien je echter terug wilt naar de oude namen eth0, eth1, enz, kan je dat op volgende wijze:

```
$ sudo nano /etc/default/grub
```

```
GRUB_CMDLINE_LINUX="net.ifnames=0 biosdevname=0"
```

```
$ sudo grub-mkconfig -o /boot/grub/grub.cfg
```

```
$ sudo nano /etc/network/interfaces
```

```
auto eth0  
iface eth0 inet static  
    address 192.168.12.12
```

```
$ sudo reboot
```



# /etc/network/interfaces

Wijziging in `/etc/network/interfaces` blijft behouden na reboot

verder: met ifconfig tijdelijke wijziging



# ifdown

take a network interface down

```
sudo ifdown eth0
```

```
sudo ifdown -a
```

```
man ifdown
```

**-a, --all**

If given to **ifup**, affect all interfaces marked **auto**. Interfaces are brought up in the order in which they are defined in /etc/network/interfaces. Combined with **--allow**, acts on all interfaces of a specified class instead. If given to **ifdown**, affect all defined interfaces. Interfaces are brought down in the order in which they are currently listed in the state file. Only interfaces defined in /etc/network/interfaces will be brought down.



# ifup

bring a network interface up

```
sudo ifup eth0
```

```
sudo ifup -a
```

```
man ifup
```

**-a, --all**

If given to **ifup**, affect all interfaces marked **auto**. Interfaces are brought up in the order in which they are defined in /etc/network/interfaces. Combined with **--allow**, acts on all interfaces of a specified class instead. If given to **ifdown**, affect all defined interfaces. Interfaces are brought down in the order in which they are currently listed in the state file. Only interfaces defined in /etc/network/interfaces will be brought down.



# ifdown && ifup

Wijziging in `/etc/network/interfaces`

nic opnieuw opstarten

```
ifdown eth0 && ifup eth0
```

rol van && :

test of 1e commando lukt?

indien ja voer 2e commando uit



# ifconfig

- informatie opvragen en wijzigingen aanbrengen
  - informatie opvragen zonder arguments: alle nic's

```
student@server1:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:82:80:50      Mac address
          inet addr:192.168.14.133      Bcast:192.168.14.255      Subnet mask
          IP address
          inet6 addr: fe80::20c:29ff:fe82:8050/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:57 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:11328 (11.3 KB)  TX bytes:4726 (4.7 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1      Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:16 errors:0 dropped:0 overruns:0 frame:0
          TX packets:16 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1184 (1.1 KB)  TX bytes:1184 (1.1 KB)
```

Broadcast address



# ifconfig

- informatie opvragen en wijzigingen aanbrengen
  - informatie opvragen met arguments: 1 specifieke nic

```
student@server1:~$ ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0c:29:82:80:50
          inet addr:192.168.14.133  Bcast:192.168.14.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe82:8050/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:57 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:11328 (11.3 KB)  TX bytes:4726 (4.7 KB)
```



# ifconfig

- gewone user heeft eventueel `/sbin` niet in `$PATH` staan:

```
normaluser@server1:~$ ifconfig eth0
Command 'ifconfig' is available in '/sbin/ifconfig'
The command could not be located because '/sbin' is not included in the PATH environment variable.
This is most likely caused by the lack of administrative privileges associated with your user account.
ifconfig: command not found
normaluser@server1:~$ /sbin/ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0c:29:82:80:50
          inet addr:192.168.14.133  Bcast:192.168.14.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe82:8050/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:57 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:11328 (11.3 KB)  TX bytes:4726 (4.7 KB)
```



- Je kan de `PATH`-variabele ook uitbreiden in de file `~/.profile`



# ifconfig -a

- Indien een interface down is, zie je hem niet met ifconfig
  - Je kan de interfaces die down zijn toch zien met
    - ifconfig -a
    - deze hebben natuurlijk geen IP-instellingen

```
student@ubserv1604:~$ ifconfig -a
ens33    Link encap:Ethernet  HWaddr 00:0c:29:e1:d9:52
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:6105 errors:0 dropped:0 overruns:0 frame:0
          TX packets:769 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:8544459 (8.5 MB)  TX bytes:61411 (61.4 KB)

lo       Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:160 errors:0 dropped:0 overruns:0 frame:0
          TX packets:160 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:11840 (11.8 KB)  TX bytes:11840 (11.8 KB)
```



# up and down

## opnieuw opstarten nic

```
ifconfig eth0 down && ifconfig eth0 up
```

huidige configuratie eth0 wordt uitgelezen en opnieuw gebruikt  
(eventueel aangebrachte wijziging via ifconfig blijft behouden)

## alternatief voor

```
ifdown eth0 && ifup eth0
```

configuratie wordt gelezen uit `/etc/network/interfaces`



# setting up IP address

## setting up MAC address

### Tijdelijke wijziging

```
student@server1:~$ ifconfig eth0 | grep 192
    inet addr:192.168.14.100 Bcast:192.168.14.255 Mask:255.255.255.0
student@server1:~$ sudo ifconfig eth0 192.168.14.101 netmask 255.255.0.0
student@server1:~$ ifconfig eth0 | grep 192
    inet addr:192.168.14.101 Bcast:192.168.255.255 Mask:255.255.0.0
student@server1:~$ sudo ifdown eth0 && sudo ifup eth0
RTNETLINK answers: No such process
student@server1:~$ ifconfig eth0 | grep 192
    inet addr:192.168.14.100 Bcast:192.168.14.255 Mask:255.255.255.0
```

```
student@server1:~$ ifconfig eth0 | grep HW
eth0      Link encap:Ethernet HWaddr 00:0c:29:82:80:50
student@server1:~$ sudo ifconfig eth0 hw ether 00:42:42:42:42:42
student@server1:~$ sudo ifdown eth0 && sudo ifup eth0
student@server1:~$ ifconfig eth0 | grep HW
eth0      Link encap:Ethernet HWaddr 00:42:42:42:42:42
```

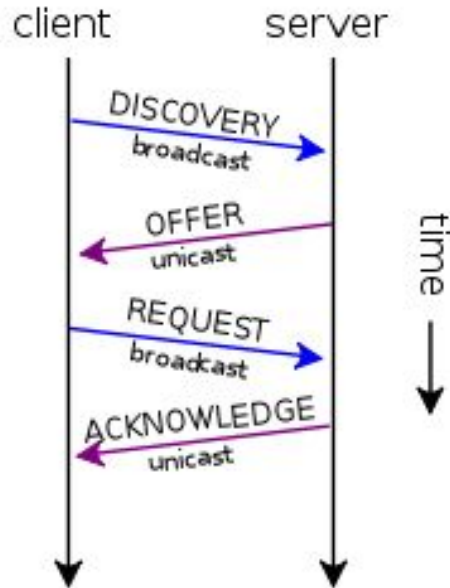
(is ook tijdelijk, maar blijft na `ifdown eth0 && ifup eth0` behouden)



# dhclient

**dhclient** = daemon op huidige OS

server = DHCP server



## DISCOVERY

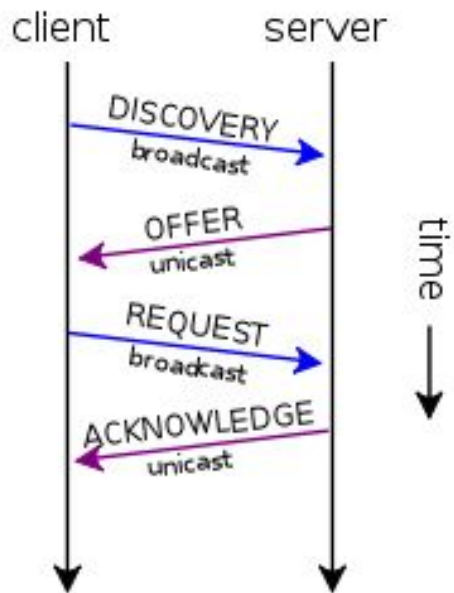
client weet niet waar DHCP-server

broadcast naar alle hosts in network

'mijn MAC address = .... gelieve mij  
een ip address te bezorgen'

# dhclient

## OFFER



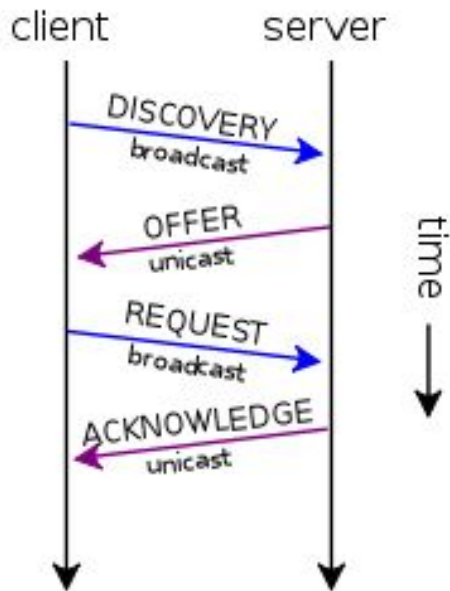
1 of meerdere DHCP servers sturen naar de client

aanbod (lease):

IP address voor client, subnet mask, lease duration en IP address van DHCP server

# dhclient

## REQUEST



client kies voor 1 van de OFFERS

doet een broadcast

'ik kies voor het OFFER van ...'

Gekozen DHCP server geeft

**ACKNOWLEDGE**

alle andere DHCP servers weten dat  
hun OFFER niet meer nodig is

# dhclient

Zie man dhclient

**Release**

```
sudo dhclient -r eth0
```

**Opnieuw lease aanvragen**

```
sudo dhclient eth0
```

(ifup zal de dhclient daemon starten)



# hostname

## Tijdelijke wijziging

Nieuwe naam zichtbaar bij het starten van een nieuwe shell

```
student@server1:~$ sudo hostname nieuwnaam
student@server1:~$ bash
student@nieuwnaam:~$ cat /etc/hostname
server1
```

/etc/hostname is niet aangepast, dus bij een reboot opnieuw oude naam



hostnames mogen bestaan uit 64 letters, cijfers, (punten en) koppeltekens, maar niet eindigen met een koppelteken.



# hostname

## Permanente wijziging

```
student@server1:~$ cat /etc/hostname  
server1  
student@server1:~$ sudo hostnamectl set-hostname nieuwenaam  
student@server1:~$ cat /etc/hostname  
nieuwenaam
```

/etc/hostname is aangepast, dus bij het starten van een nieuwe shell en het herstarten van de PC in de toekomst blijft de nieuwe naam behouden



# hostname

Aanpassen van /etc/hosts voor name-resolving (voor sudo)

- sudo doet voor ieder commando een name-resolving voor de hostname
  - Daarom is het belangrijk dat je de file /etc/hosts ook aanpast
    - anders heb je een lange timeout voordat een sudo commando wordt uitgevoerd

```
student@server1:~$ cat /etc/hostname
server1
student@server1:~$ cat /etc/hosts
127.0.0.1    localhost
127.0.1.1    server1

# The following lines are desirable for IPv6 capable hosts
::1        localhost ip6-localhost ip6-loopback
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters
```



# arp

```
Terminal
ARP(8)                                Linux Programmer's Manual                                ARP(8)

NAME
    arp - manipulate the system ARP cache

DESCRIPTION
    Arp manipulates or displays the kernel's IPv4 network neighbour cache.
    It can add entries to the table, delete one or display the current content.

    ARP stands for Address Resolution Protocol, which is used to find the
    media access control address of a network neighbour for a given IPv4
    Address.
```



# arp

Toon alle entries

```
student@server1:~$ arp -a
? (192.168.14.2) at 00:50:56:ff:52:ca [ether] on eth0
? (192.168.14.254) at 00:50:56:f5:76:3e [ether] on eth0
```

```
student@destkop1:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:1d:cb:aa
          inet addr:192.168.14.134  Bcast:192.168.14.255  Mask:255.255.255.0
          inet6 addr: fe80::500:29:29:55:61:cb:aa/64 Scope:link
```

```
student@server1:~$ ping 192.168.14.134
PING 192.168.14.134 (192.168.14.134) 56(84) bytes of data.
64 bytes from 192.168.14.134: icmp_seq=1 ttl=64 time=0.453 ms
```

```
student@server1:~$ arp -a
? (192.168.14.134) at 00:0c:29:1d:cb:aa [ether] on eth0
? (192.168.14.2) at 00:50:56:ff:52:ca [ether] on eth0
? (192.168.14.254) at 00:50:56:f5:76:3e [ether] on eth0
```

Delete entry

```
student@server1:~$ sudo arp -d 192.168.14.134
student@server1:~$ arp -a
? (192.168.14.134) at <incomplete> on eth0
? (192.168.14.2) at 00:50:56:ff:52:ca [ether] on eth0
? (192.168.14.254) at 00:50:56:f5:76:3e [ether] on eth0
```

(of met hostname)



# route

```
Terminal
ROUTE(8) Linux Programmer's Manual ROUTE(8)

NAME
    route - show / manipulate the IP routing table

DESCRIPTION
    Route manipulates the kernel's IP routing tables. Its primary use is to set up static routes to specific hosts or networks via an interface after it has been configured with the ifconfig(8) program.

    When the add or del options are used, route modifies the routing tables. Without these options, route displays the current contents of the routing tables.
```



# route

```
student@server1:~$ route
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
default          192.168.14.2   0.0.0.0         UG      0      0        0 eth0
192.168.14.0     *              255.255.255.0   U        0      0        0 eth0
192.168.14.0     *              255.255.255.0   U        0      0        0 eth1
```

of

```
student@server1:~$ netstat -r
Kernel IP routing table
Destination      Gateway         Genmask         Flags  MSS Window  irtt Iface
default          192.168.14.2   0.0.0.0         UG      0  0        0 eth0
192.168.14.0     *              255.255.255.0   U        0  0        0 eth0
192.168.14.0     *              255.255.255.0   U        0  0        0 eth1
```

Tijdelijk wijzigingen routing: (vb. default gateway)

```
sudo route add default gw 192.168.14.xx
```

(man route)



# ping

Met ping wordt vaak de TCP/IP configuratie getest.

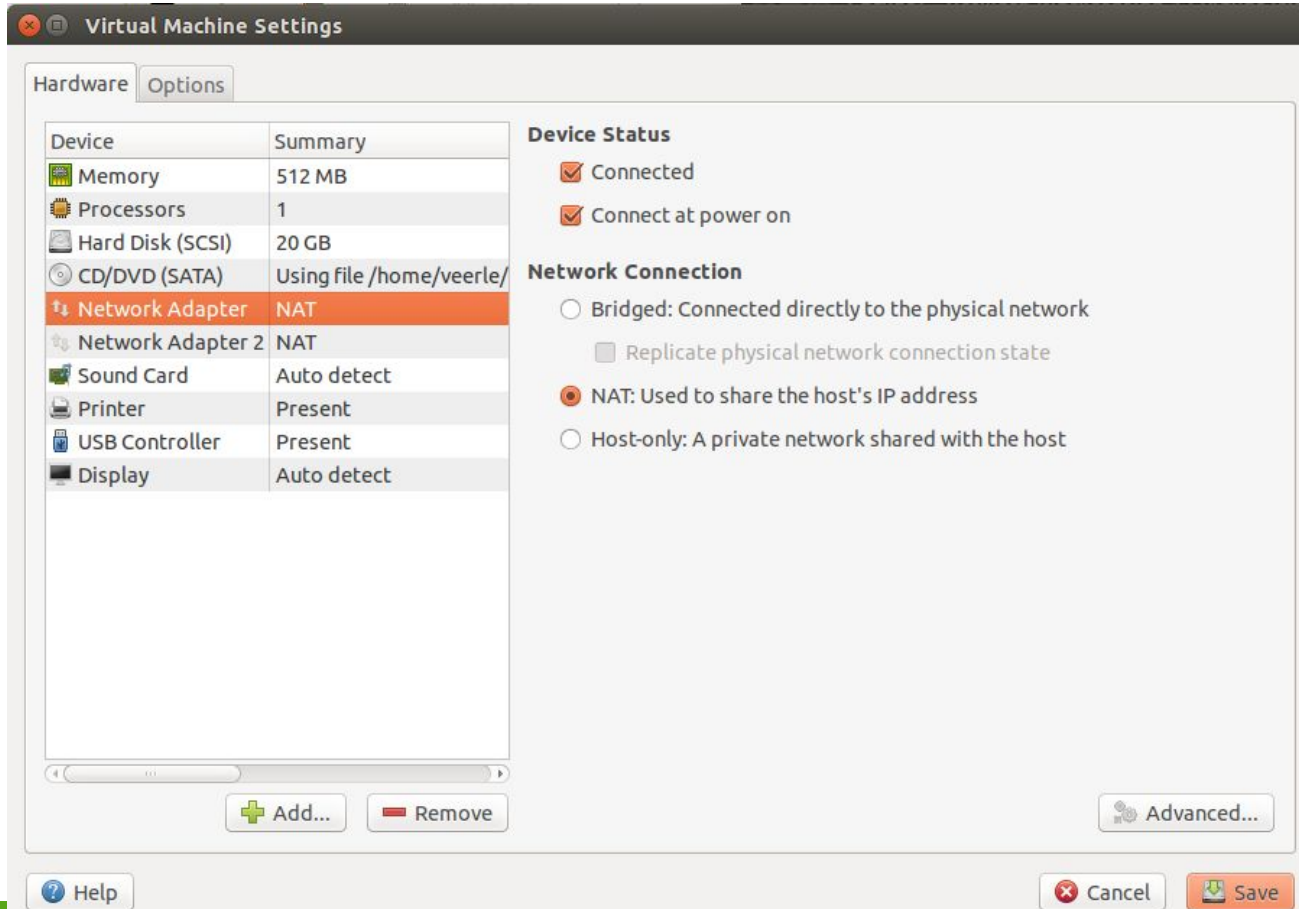
(ook traceroute, dig)

Ctrl-C

```
student@server1:~$ ping 192.168.14.134
PING 192.168.14.134 (192.168.14.134) 56(84) bytes of data.
64 bytes from 192.168.14.134: icmp_seq=1 ttl=64 time=0.873 ms
64 bytes from 192.168.14.134: icmp_seq=2 ttl=64 time=0.190 ms
64 bytes from 192.168.14.134: icmp_seq=3 ttl=64 time=1.95 ms
^C
--- 192.168.14.134 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.190/1.005/1.953/0.726 ms
```



# VMware: NAT





# VMware: NAT - host

interface vmnet8

```
root@kali:~$ ifconfig vmnet8
vmnet8    Link encap:Ethernet  HWaddr 00:50:56:c0:00:08
          inet addr:192.168.14.1  Bcast:192.168.14.255  Mask:255.255.255.0
          inet6 addr: fe80::250:56ff:fec0:8/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

NAT router en DNS Server

ip = 192.168.14.2

DHCP Server

ip = 192.168.14.254



# VMware: NAT - VM

```
student@server1:~$ ifconfig eth0 | grep inet
    inet addr:192.168.14.100 Bcast:192.168.14.255 Mask:255.255.255.0
    inet6 addr: fe80::20c:29ff:fe82:8050/64 Scope:Link
student@server1:~$ cat /var/lib/dhcp/dhclient.leases | grep lease -A15
lease {
    interface "eth1";
    fixed-address 192.168.14.136;
    option subnet-mask 255.255.255.0;
    option routers 192.168.14.2;
    option dhcp-lease-time 1800;
    option dhcp-message-type 5;
    option domain-name-servers 192.168.14.2;
    option dhcp-server-identifier 192.168.14.254;
    option broadcast-address 192.168.14.255;
    option netbios-name-servers 192.168.14.2;
    option domain-name "localdomain";
    renew 1 2015/10/12 18:13:38;
    rebind 1 2015/10/12 18:28:35;
    expire 1 2015/10/12 18:32:20;
}
```



# VMware: NAT - VM

```
student@server1:~$ route
Kernel IP routing table
Destination      Gateway          Genmask
default          192.168.14.2    0.0.0.0
192.168.14.0     *               255.255.255.0
192.168.14.0     *               255.255.255.0
```

```
student@server1:~$ dig www.google.be

;<>> DiG 9.9.5-3-Ubuntu <>> www.google.be
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59758
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 5

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; MBZ: 0005 , udp: 4096
;; QUESTION SECTION:
;www.google.be.
IN      A

;; ANSWER SECTION:
www.google.be.      5      IN      A      64.233.166.94

;; AUTHORITY SECTION:
google.be.          5      IN      NS      ns3.google.com.
google.be.          5      IN      NS      ns4.google.com.
google.be.          5      IN      NS      ns1.google.com.
google.be.          5      IN      NS      ns2.google.com.

;; ADDITIONAL SECTION:
ns1.google.com.     5      IN      A      216.239.32.10
ns2.google.com.     5      IN      A      216.239.34.10
ns3.google.com.     5      IN      A      216.239.36.10
ns4.google.com.     5      IN      A      216.239.38.10

;; Query time: 36 msec
;; SERVER: 192.168.14.2#53(192.168.14.2)
;; WHEN: Mon Oct 12 20:51:52 CEST 2015
;; MSG SIZE rcvd: 204
```

DNS - poort 53



# ifconfig vs ip

## COMPARING NET-TOOLS VS. IPRROUTE PACKAGE COMMANDS

### NET-TOOLS COMMANDS

### IPROUTE COMMANDS

arp -a

ip neigh

arp -v

ip -s neigh

arp -s 192.168.1.1 1:2:3:4:5:6

ip neigh add 192.168.1.1 lladdr 1:2:3:4:5:6 dev eth1

arp -i eth1 -d 192.168.1.1

ip neigh del 192.168.1.1 dev eth1

ifconfig -a

ip addr

ifconfig eth0 down

ip link set eth0 down

ifconfig eth0 up

ip link set eth0 up

ifconfig eth0 192.168.1.1

ip addr add 192.168.1.1/24 dev eth0

ifconfig eth0 netmask 255.255.255.0

ip addr add 192.168.1.1/24 dev eth0

ifconfig eth0 mtu 9000

ip link set eth0 mtu 9000

ifconfig eth0:0 192.168.1.2

ip addr add 192.168.1.2/24 dev eth0

netstat

ss

netstat -neopa

ss -neopa

netstat -g

ip maddr

route

ip route

route add -net 192.168.1.0 netmask 255.255.255.0 dev eth0

ip route add 192.168.1.0/24 dev eth0

route add default gw 192.168.1.1

ip route add default via 192.168.1.1

