



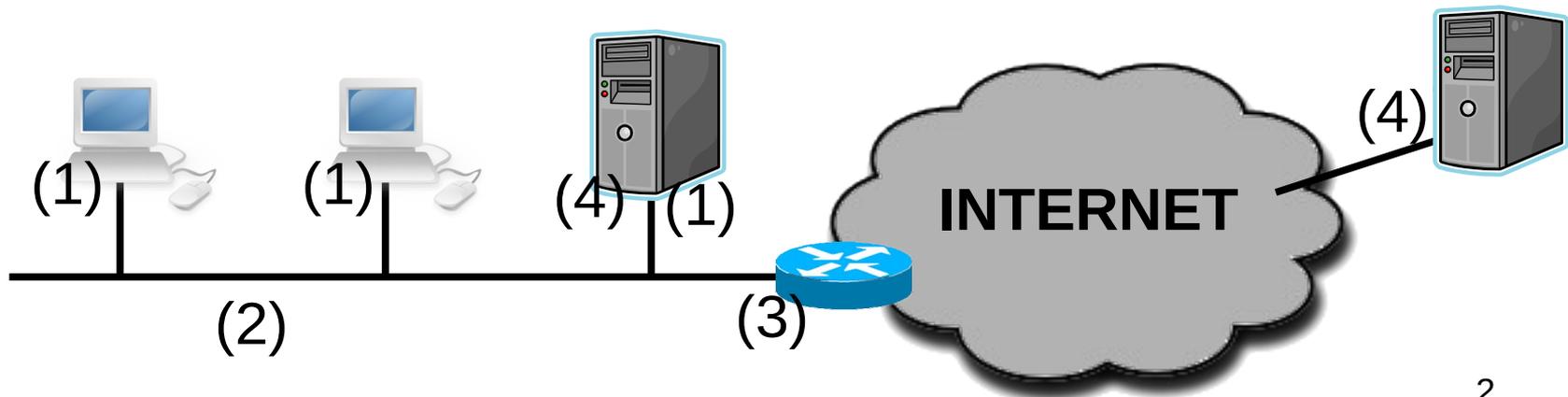
Configurazione delle interfacce di rete

A.A. 2018/2019

Walter Cerroni

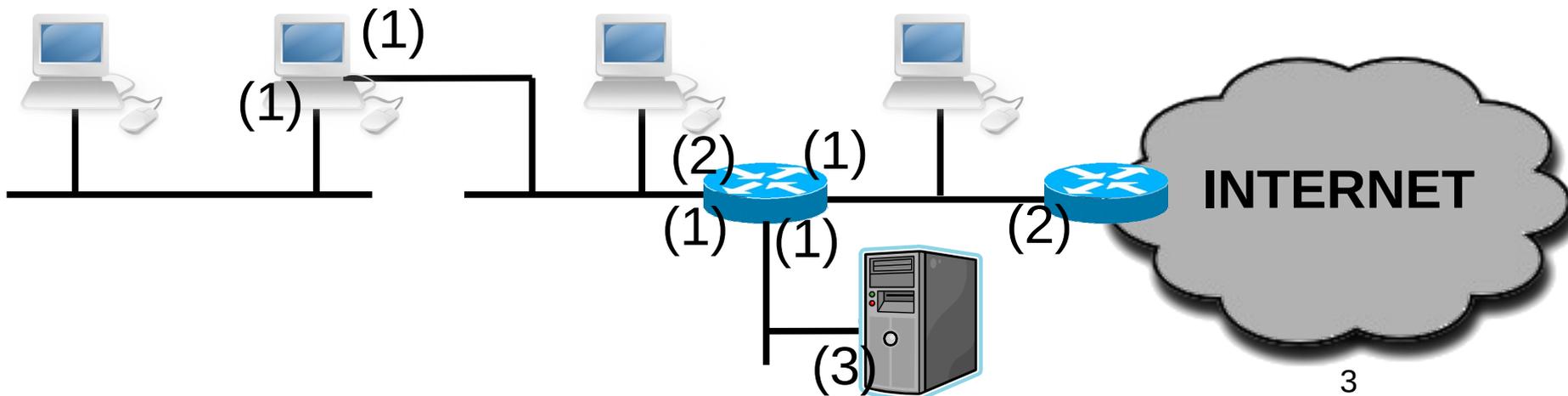
Configurazione interfacce di rete di un host

- Per configurare un host con un'unica interfaccia di rete occorre specificare:
 - (1) indirizzo IP da assegnare all'interfaccia
 - (2) netmask per stabilire le dimensioni della sua rete IP
 - (3) indirizzo IP del router a cui inoltrare i pacchetti diretti all'esterno della rete IP (*default gateway*)
 - (4) indirizzo IP dei server DNS da utilizzare per tradurre i nomi di dominio (es. `www.unibo.it`) in indirizzi IP

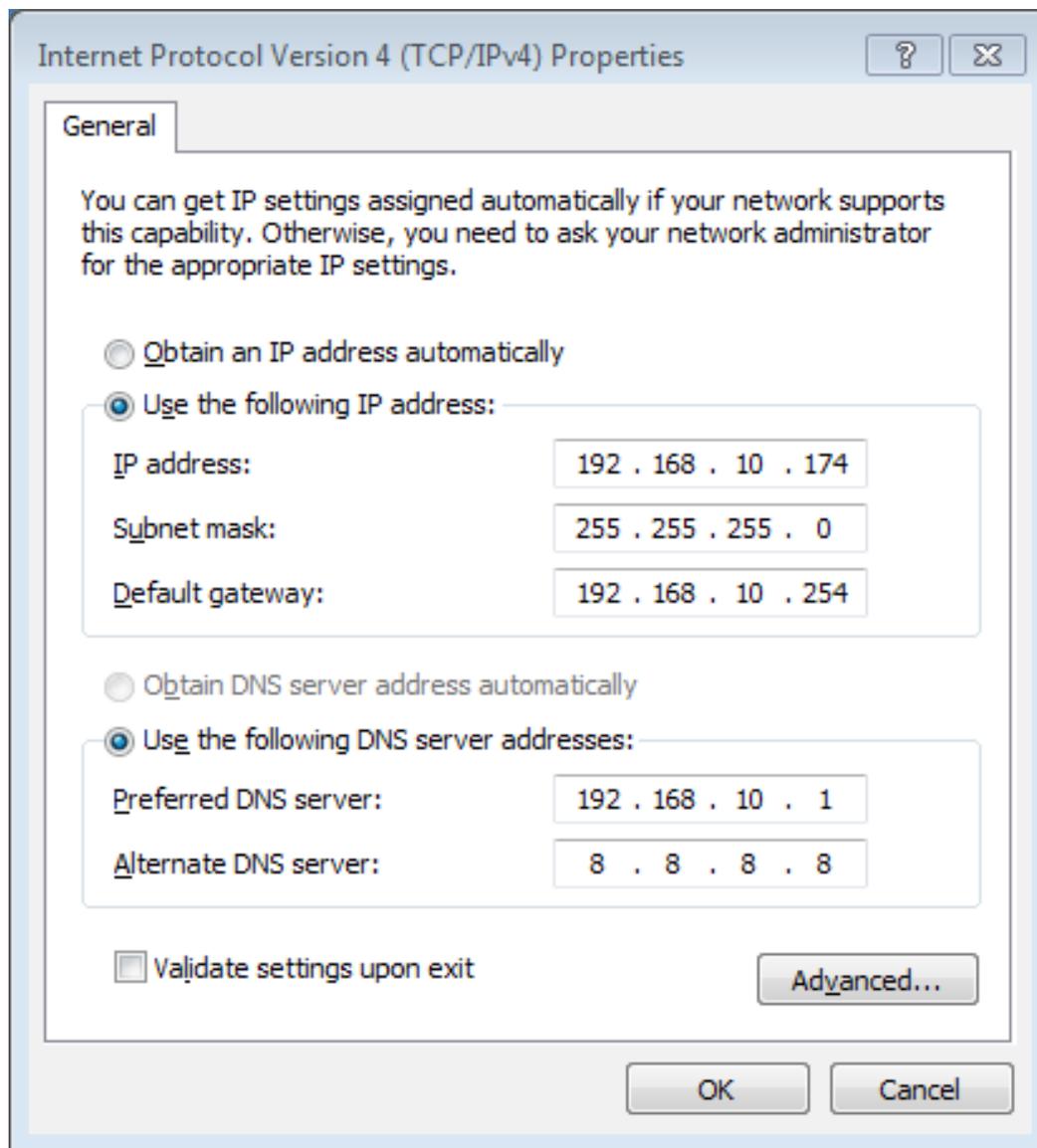


Configurazione interfacce di rete di un host

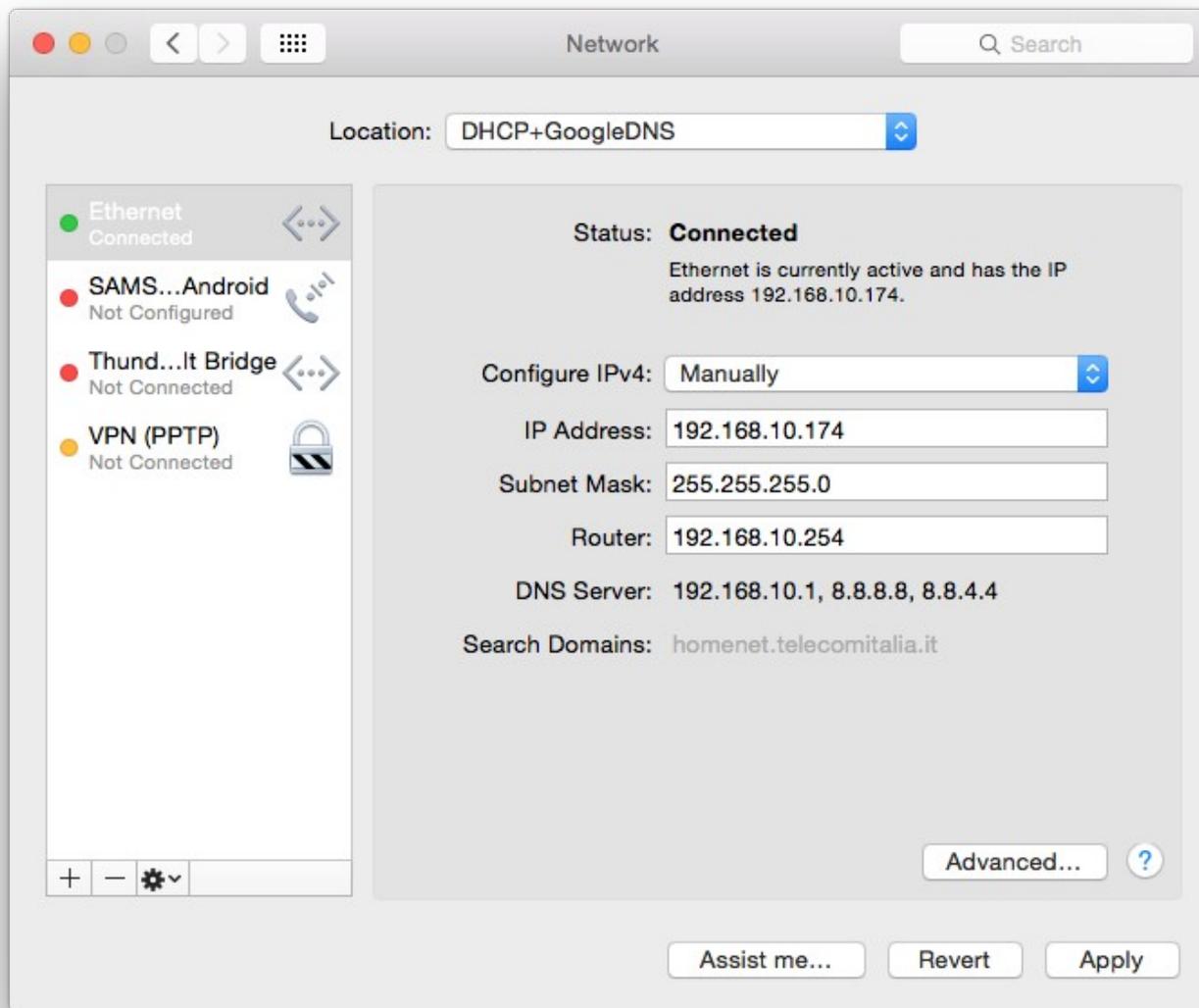
- Per configurare un host con interfacce di rete multiple o un router occorre specificare:
 - (1) indirizzo IP e netmask da assegnare a ciascuna interfaccia
 - (2) indirizzo IP del default gateway (unico)
 - (3) indirizzo IP dei server DNS
- Per configurare un router, (2) e (3) non sono strettamente necessari



Configurazione interfacce di rete (Windows)



Configurazione interfacce di rete (Mac)



Configurazione interfacce di rete (Linux NM)

The screenshot shows the NetworkManager configuration window for the interface 'DEI Bologna'. The window is divided into several sections:

- Left sidebar:** A vertical list of tabs: 'Details', 'Security', 'Identity', 'IPv4' (selected), 'IPv6', and 'Reset'.
- IPv4 section:** Located at the top right, it features a toggle switch set to 'ON'.
- Addresses section:** Below the IPv4 toggle, it shows a dropdown menu set to 'Manual'. It contains three input fields: 'Address' (192.168.10.174), 'Netmask' (255.255.255.0), and 'Gateway' (192.168.10.254). Each field has a trash icon to its right. A '+' button is at the bottom right of this section.
- DNS section:** Below the addresses, it features a toggle switch set to 'OFF'. It contains two input fields for 'Server' (192.168.10.1 and 192.168.10.2), each with a trash icon to its right. A '+' button is at the bottom right of this section.
- Bottom:** Two buttons labeled 'Cancel' and 'Apply'.

Configurazione interfacce di rete (Linux CLI)

Il comando **ifconfig** elenca e descrive le interfacce di rete attive (**ifconfig -a** per elencare anche quelle non attive)

```
[labuser@netlab07 ~]$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0E:0C:69:18:5C
          inet addr:10.0.0.13  Bcast:10.0.0.15  Mask:255.255.255.252
          inet6 addr: fe80::20e:cff:fe69:185c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:11 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:678 (678.0 b)

eth1      Link encap:Ethernet  HWaddr 00:07:E9:F6:34:DF
          inet addr:192.168.8.7  Bcast:192.168.8.255  Mask:255.255.255.0
          inet6 addr: fe80::207:e9ff:fef6:34df/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:8077 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3139 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:546911 (534.0 KiB)  TX bytes:314598 (307.2 KiB)

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:16436  Metric:1
          RX packets:28 errors:0 dropped:0 overruns:0 frame:0
          TX packets:28 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1560 (1.5 KiB)  TX bytes:1560 (1.5 KiB)
```

Configurazione interfacce di rete (Linux CLI)

Il comando **route -n** mostra la tabella di routing

Il file **/etc/resolv.conf** contiene gli indirizzi dei server DNS

```
[labuser@netlab07 ~]$ route -n
Kernel IP routing table
Destination      Gateway          Genmask         Flags Metric Ref    Use Iface
10.0.0.12        0.0.0.0         255.255.255.252 U        0      0      0 eth0
192.168.8.0      0.0.0.0         255.255.255.0   U        0      0      0 eth1
0.0.0.0          192.168.8.254   0.0.0.0         UG       0      0      0 eth1
[labuser@netlab07 ~]$
[labuser@netlab07 ~]$
[labuser@netlab07 ~]$ cat /etc/resolv.conf
nameserver 137.204.78.3
nameserver 137.204.78.2
[labuser@netlab07 ~]$
```

Configurazione interfacce di rete (Linux CLI)

Il comando **ifconfig** permette anche di modificare in tempo reale (ma temporaneamente) la configurazione corrente delle interfacce di rete

```
[labuser@netlab07 ~]$ ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0E:0C:69:18:5C
          inet addr:10.0.0.13  Bcast:10.0.0.15  Mask:255.255.255.252
          inet6 addr: fe80::20e:cff:fe69:185c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:33 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:2034 (1.9 KiB)

[labuser@netlab07 ~]$ sudo ifconfig eth0 10.100.100.131/25
Password:
[labuser@netlab07 ~]$ ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0E:0C:69:18:5C
          inet addr:10.100.100.131  Bcast:10.100.100.255  Mask:255.255.255.128
          inet6 addr: fe80::20e:cff:fe69:185c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:33 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:2034 (1.9 KiB)
```

Configurazione interfacce di rete (Linux CLI)

Il comando **ifconfig** aggiorna automaticamente il contenuto della tabella di routing relativo all'interfaccia appena modificata

```
[labuser@netlab07 ~]$ sudo ifconfig eth0 10.100.100.131/25
Password:
[labuser@netlab07 ~]$ ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0E:0C:69:18:5C
          inet addr:10.100.100.131  Bcast:10.100.100.255  Mask:255.255.255.128
          inet6 addr: fe80::20e:cff:fe69:185c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:33 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:2034 (1.9 KiB)

[labuser@netlab07 ~]$ route -n
Kernel IP routing table
Destination        Gateway            Genmask           Flags Metric Ref    Use Iface
10.100.100.128    0.0.0.0           255.255.255.128  U         0      0      0 eth0
192.168.8.0       0.0.0.0           255.255.255.0    U         0      0      0 eth1
0.0.0.0           192.168.8.254    0.0.0.0          UG        0      0      0 eth1
[labuser@netlab07 ~]$
```

Sintassi del comando ifconfig

- **ifconfig** <interface> <address> **netmask** <mask>
broadcast <broadcast-address>
assegna un indirizzo ad una interfaccia specificando la netmask e l'indirizzo di broadcast
<interface> è l'interfaccia da configurare (es. eth0, eth1, ...)
<address> è l'indirizzo IP da assegnare (es. 192.168.8.27)
<mask> è la netmask da associare all'IP (es. 255.255.255.0)
<broadcast-address> è l'indirizzo IP broadcast della rete a cui appartiene l'interfaccia (es. 192.168.8.255)
- netmask e broadcast vengono calcolati automaticamente se si specifica l'indirizzo in notazione CIDR:
ifconfig <interface> <address>/<prefix-length>
- **ifconfig** <interface> **up**
attiva l'interfaccia specificata
- **ifconfig** <interface> **down**
disattiva l'interfaccia specificata

Configurazione interfacce di rete (Linux CLI)

In alternativa, i comandi **ip addr** e **ip route** svolgono le stesse funzioni dei comandi **ifconfig** e **route**

```
[labuser@netlab07 ~]$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:0e:0c:69:18:5c brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.13/30 brd 10.0.0.15 scope global eth0
    inet6 fe80::20e:cff:fe69:185c/64 scope link
        valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:07:e9:f6:34:df brd ff:ff:ff:ff:ff:ff
    inet 192.168.8.7/24 brd 192.168.8.255 scope global eth1
    inet6 fe80::207:e9ff:fef6:34df/64 scope link
        valid_lft forever preferred_lft forever
4: sit0: <NOARP> mtu 1480 qdisc noop
    link/sit 0.0.0.0 brd 0.0.0.0
[labuser@netlab07 ~]$ ip route show
10.0.0.12/30 dev eth0 proto kernel scope link src 10.0.0.13
192.168.8.0/24 dev eth1 proto kernel scope link src 192.168.8.7
default via 192.168.8.254 dev eth1
```

Sintassi del comando ip

- **ip addr add <address>/<prefix-length> dev <interface> broadcast +**

aggiunge un indirizzo ad un'interfaccia, specificando l'indirizzo in notazione CIDR e calcolando automaticamente il broadcast

- per sostituire un indirizzo assegnato, occorre prima eliminare il precedente:

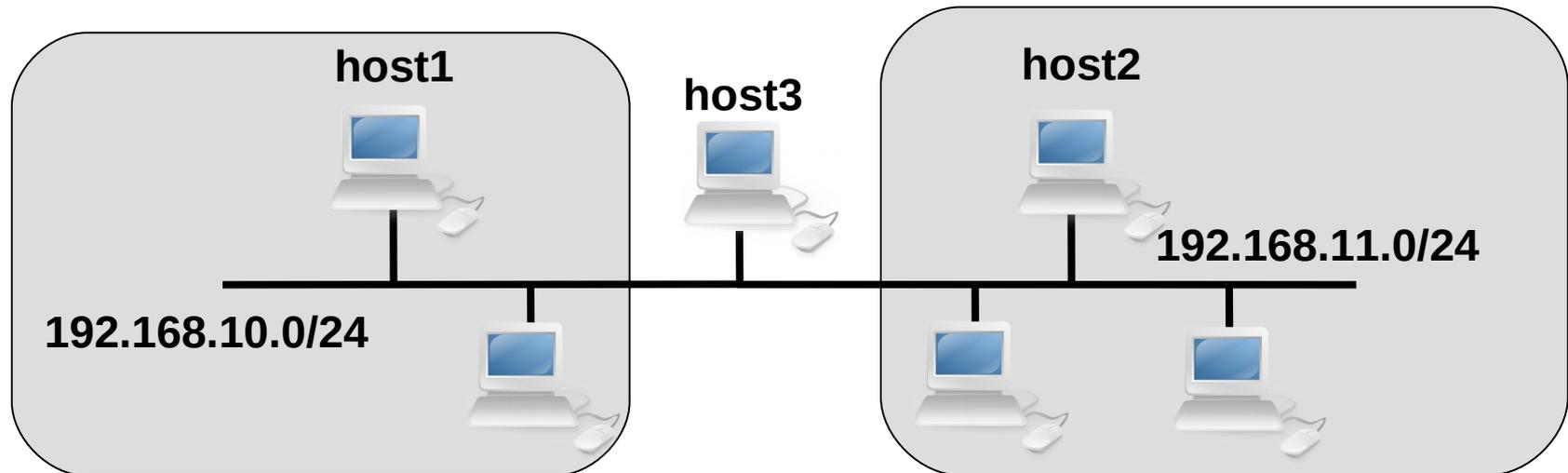
ip addr del <address>/<prefix-length> dev <interface>

- **ip link set <interface> up**
attiva l'interfaccia specificata
- **ip link set <interface> down**
disattiva l'interfaccia specificata

Assegnare indirizzi multipli ad un'interfaccia

La stessa interfaccia di rete può avere più di un indirizzo IP utile se l'interfaccia è connessa a reti IP multiple presenti sulla stessa rete fisica

1. usando **alias IP**
2. usando **indirizzi IP secondari**



Assegnare indirizzi multipli ad un'interfaccia

1. Usando alias IP (ifconfig)

```
[labuser@netlab07 ~]$ sudo ifconfig eth0:alias1 10.200.200.33/24
[labuser@netlab07 ~]$ sudo ifconfig eth0:alias2 10.255.255.2/29
[labuser@netlab07 ~]$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0E:0C:69:18:5C
          inet addr:10.100.100.131  Bcast:10.100.100.255  Mask:255.255.255.128
          inet6 addr: fe80::20e:cff:fe69:185c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:33 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:2034 (1.9 KiB)

eth0:alias1 Link encap:Ethernet  HWaddr 00:0E:0C:69:18:5C
          inet addr:10.200.200.33  Bcast:10.200.200.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

eth0:alias2 Link encap:Ethernet  HWaddr 00:0E:0C:69:18:5C
          inet addr:10.255.255.2  Bcast:10.255.255.7  Mask:255.255.255.248
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

Assegnare indirizzi multipli ad un'interfaccia

2. Usando indirizzi IP secondari (ip addr)

```
[labuser@netlab07 ~]$ sudo ip addr add 10.200.200.33/24 dev eth0
[labuser@netlab07 ~]$ sudo ip addr add 10.255.255.2/29 dev eth0
[labuser@netlab07 ~]$ ip addr show dev eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:0e:0c:69:18:5c brd ff:ff:ff:ff:ff:ff
    inet 10.100.100.131/25 brd 10.100.100.255 scope global eth0
    inet 10.200.200.33/24 scope global eth0
    inet 10.255.255.2/29 scope global eth0
    inet6 fe80::20e:cff:fe69:185c/64 scope link
        valid_lft forever preferred_lft forever
[labuser@netlab07 ~]$ route -n
Kernel IP routing table
Destination        Gateway            Genmask           Flags Metric Ref    Use Iface
10.255.255.0       0.0.0.0           255.255.255.248  U         0      0      0 eth0
10.100.100.128    0.0.0.0           255.255.255.128  U         0      0      0 eth0
10.200.200.0      0.0.0.0           255.255.255.0    U         0      0      0 eth0
192.168.8.0       0.0.0.0           255.255.255.0    U         0      0      0 eth1
0.0.0.0           192.168.8.254    0.0.0.0          UG        0      0      0 eth1
[labuser@netlab07 ~]$
```

Gli indirizzi secondari non vengono visualizzati da **ifconfig**

ifconfig vs. ip addr

ifconfig:

- Mostra solo le interfacce attive
- Sostituisce il precedente indirizzo, se già assegnato

ip addr:

- Mostra tutte le interfacce (sia in stato *UP* che *DOWN*)
- Aggiunge all'interfaccia un indirizzo IP secondario (ogni indirizzo precedentemente assegnato va eliminato esplicitamente)

Interfacce wireless – Comando iwconfig

iwconfig <interface> [options]

configura i parametri per le interfacce Wi-Fi (IEEE 802.11)

<interface> è l'interfaccia da configurare (es. eth1, ath0, wlan0, ...)

essid X: sceglie X come SSID della rete wireless a cui connettersi

mode M: imposta la modalità di funzionamento

M = Ad-Hoc, Managed, Monitor, Repeater, Master, Auto

channel C: imposta il canale (o la frequenza) su cui operare

sens S: imposta la soglia di segnale al di sotto della quale si preferisce scartare i dati ricevuti

ap A: specifica l'indirizzo MAC dell'access point a cui agganciarsi

rate R: specifica la bit-rate a cui trasmettere

key K: imposta la chiave per la cifratura/autenticazione WEP

txpower T: imposta la potenza di trasmissione

rts RT, **frag FT**, **power P**, **retry R**: configura i dettagli del protocollo di accesso al mezzo radio

Interfacce wireless – Comando iwconfig

```
[walter@deisnet99 ~]$ iwconfig
enol          no wireless extensions.

lo            no wireless extensions.

wlp3s0        IEEE 802.11abgn  ESSID:"MyWiFi"
              Mode:Managed  Frequency:5.54 GHz  Access Point: 32:91:8F:47:9F:71
              Bit Rate=216 Mb/s   Tx-Power=15 dBm
              Retry short limit:7   RTS thr:off   Fragment thr:off
              Power Management:off
              Link Quality=50/70   Signal level=-60 dBm
              Rx invalid nwid:0    Rx invalid crypt:0    Rx invalid frag:0
              Tx excessive retries:0 Invalid misc:28   Missed beacon:0

[walter@deisnet99 ~]$ █
```

- L'accesso a reti Wi-Fi sicure di tipo WPA e WPA2 si configura separatamente:
 - configurazione grafica tramite NetworkManager
 - configurazione CLI tramite **wpa_supplicant**

Rimuovere la configurazione IP corrente

Rimuovere l'indirizzo IP

```
[labuser@netlab07 ~]$ sudo ifconfig eth0 0
```

```
[labuser@netlab07 ~]$ sudo ip addr flush dev eth0
```

Rimuovere alias e indirizzi IP secondari:

```
[labuser@netlab07 ~]$ sudo ifconfig eth0:alias1 down  
[labuser@netlab07 ~]$ sudo ifconfig eth0:alias2 down
```

```
[labuser@netlab07 ~]$ sudo ip addr del 10.200.200.33/24 dev eth0  
[labuser@netlab07 ~]$ sudo ip addr del 10.255.255.2/29 dev eth0
```

Disattivare un'interfaccia:

```
[labuser@netlab07 ~]$ sudo ifconfig eth0 down
```

```
[labuser@netlab07 ~]$ sudo ip link set eth0 down
```

File di configurazione di rete

- Per specificare i parametri di rete da attivare all'avvio, si usano specifici script di inizializzazione/configurazione
- Per distribuzioni RedHat/CentOS/Fedora (e simili):
 - configurazione **`/etc/sysconfig/network-scripts/ifcfg-eth0`**
`/etc/sysconfig/network-scripts/ifcfg-eth1`
 - inizializzazione tramite comando di gestione servizi
`systemctl <action> <service>`
 - il comando **`systemctl`** legge i file di configurazione di un dato servizio ed esegue su quel servizio l'azione specificata
 - è possibile attivare/disattivare/reimpostare/verificare lo stato delle interfacce di rete secondo i parametri specificati nei file con
`systemctl [start | stop | restart | status] network`

File di configurazione di rete

- Esempio di file di configurazione per distribuzioni RedHat/CentOS/Fedora (**ifcfg-eth0**)

DEVICE=eth0

BOOTPROTO=none

HWADDR=00:0E:0C:69:17:40

IPADDR=192.168.11.172

PREFIX=24

GATEWAY=192.168.11.254

DNS1=137.204.78.17

DNS2=8.8.8.8

ONBOOT=yes

NOZEROCONF=yes

File di configurazione di rete

- Per specificare i parametri di rete da attivare all'avvio, si usano specifici script di inizializzazione/configurazione
- Per distribuzioni Debian/Ubuntu (e simili):
 - configurazione **/etc/network/interfaces**
 - inizializzazione **/etc/init.d/networking**
 - è possibile attivare/disattivare/reimpostare/verificare lo stato delle interfacce di rete secondo i parametri specificati negli script con **service networking [start | stop | restart]**

File di configurazione di rete

- Esempio di file di configurazione per distribuzioni Debian/Ubuntu (**interfaces**)

```
auto lo
```

```
iface lo inet loopback
```

```
auto eth0
```

```
iface eth0 inet static
```

```
    address 192.168.11.172
```

```
    netmask 255.255.255.0
```

```
    gateway 192.168.11.254
```

```
    dns-nameservers 137.204.78.17 8.8.8.8
```

```
auto eth1
```

```
iface eth1 inet static
```

```
    address 10.15.0.2
```

```
    netmask 255.255.255.252
```

DHCP – RFC 2131

Dynamic Host Configuration Protocol

Configurazione automatica e dinamica di

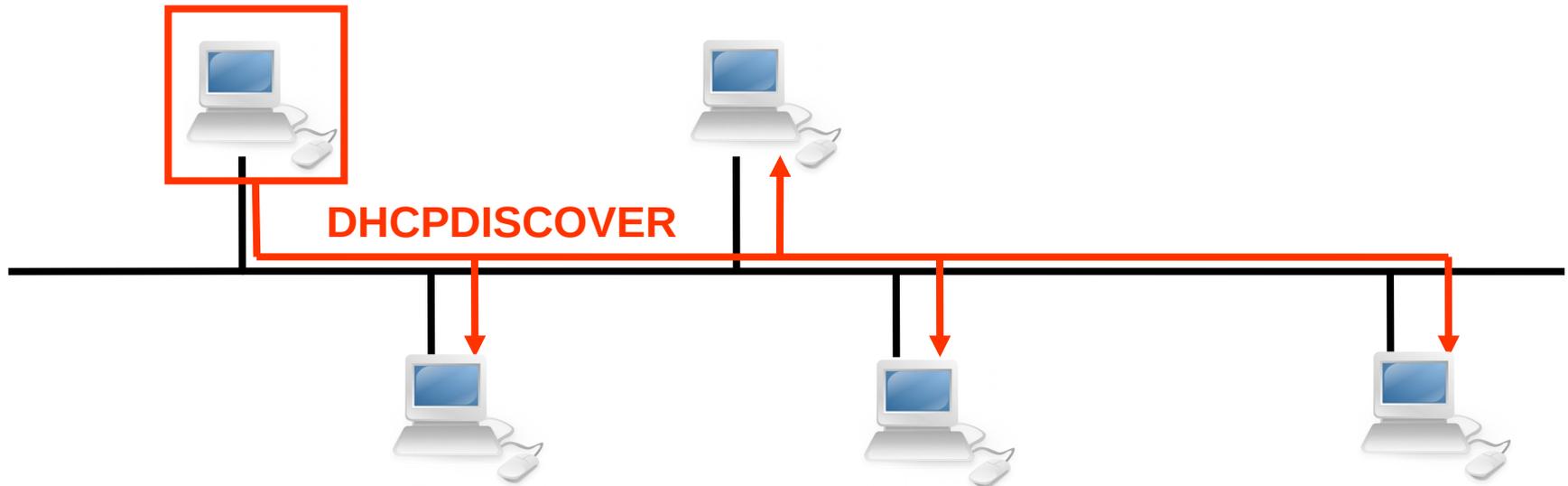
- Indirizzo IP
- Netmask
- Broadcast
- Host name
- Default gateway
- Server DNS

Server su porta 67 UDP

Client su porta 68 UDP

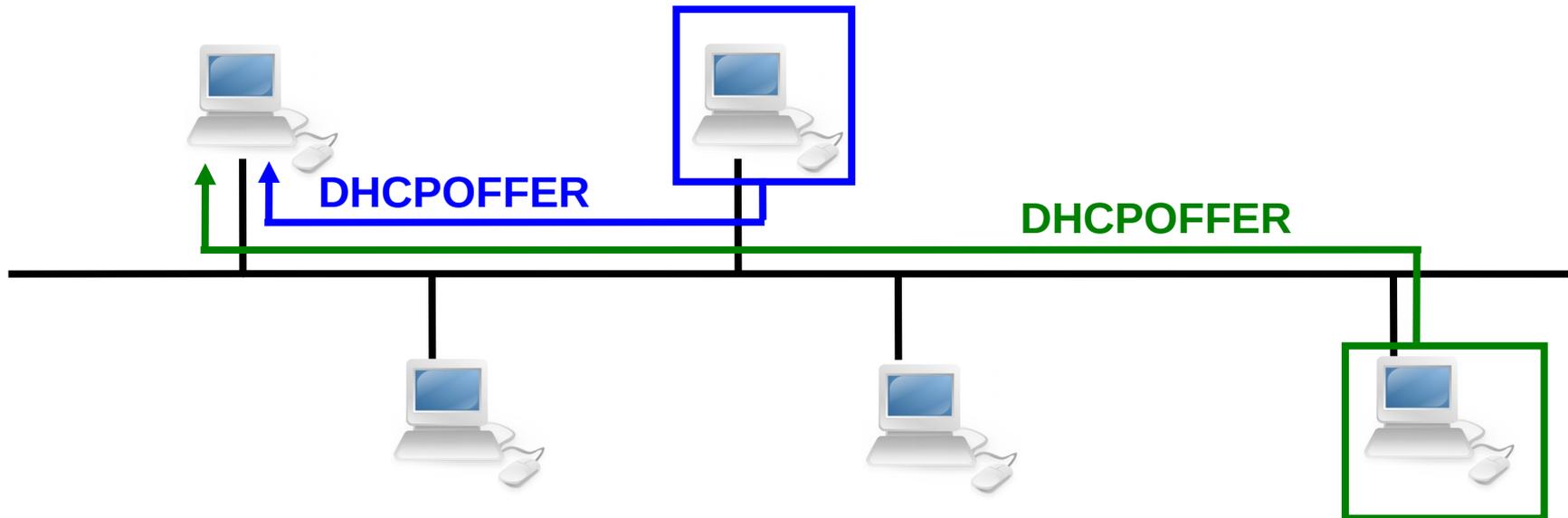
DHCP – 1

- Quando un host attiva l'interfaccia di rete, invia in modalità broadcast un messaggio **DHCPDISCOVER** in cerca di un server DHCP



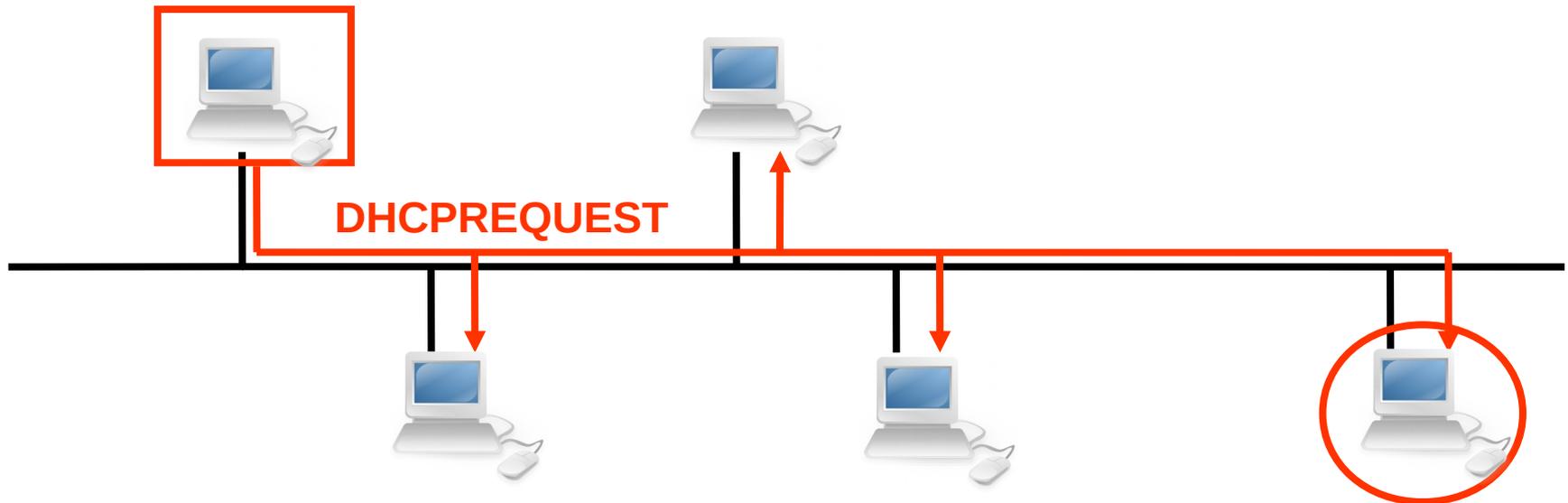
DHCP – 2

- Ciascun server DHCP presente risponde all'host con un messaggio **DHCPOFFER** con cui propone un indirizzo IP



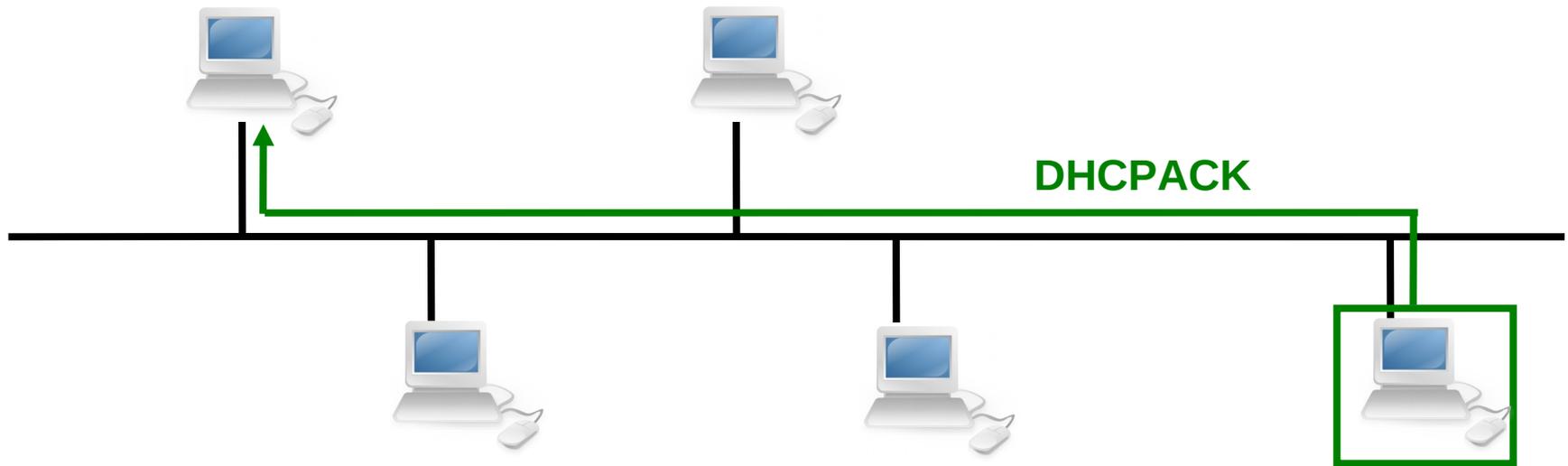
DHCP – 3

- L'host accetta una delle offerte proposte dai server e manda un messaggio **DHCPREQUEST** in cui richiede la configurazione, specificando il server



DHCP – 4

- Il server DHCP risponde all'host con un messaggio **DHCPACK** specificando tutti i parametri di configurazione



DHCP leasing

- DHCP permette l'allocazione dinamica degli indirizzi IP
- Host diversi possono utilizzare lo stesso indirizzo in tempi diversi
- L'allocazione deve essere limitata ad un intervallo temporale stabilito dal server, eventualmente rinnovabile
- Quando un host viene spento o disconnesso deve rilasciare l'indirizzo e renderlo disponibile per una nuova allocazione
- Il rilascio può essere esplicito (**DHCPRELEASE**) o avvenire allo scadere del periodo di lease
- Il tempo di lease è uno dei parametri forniti dal server nel DHCPOFFER

Linux come server DHCP

- Configurazione in **`/etc/dhcp/dhcpd.conf`**
 - in genere solo root ha i diritti di scrittura
 - documentazione accessibile con **`man dhcpd.conf`**
- Il demone utilizzato è `/usr/sbin/dhcpd`
- Da avviare come root con il comando **`systemctl`**
`systemctl [start | stop | restart | status] dhcpd`
- Registro degli assegnamenti in
`/var/lib/dhcpd/dhcpd.leases`

dhcpcd.conf – configurazione di una subnet

Esempio di file di configurazione per dhcpcd

```
subnet 192.168.100.0 netmask 255.255.255.0 {  
    range 192.168.100.64 192.168.100.127;  
    range 192.168.100.200 192.168.100.210;  
    default-lease-time 3600;  
    max-lease-time 10800;  
    option subnet-mask 255.255.255.0;  
    option broadcast-address 192.168.100.255;  
    option routers 192.168.100.254;  
    option domain-name-servers 8.8.8.8;  
}
```

almeno una delle
interfacce del sistema
deve appartenere alla
subnet specificata

insiemi di indirizzi usati
per il servizio DHCP

parametri comuni a
tutta la subnet

dhcpcd.conf – configurazione di host specifici

```
# Esempio di file di configurazione per dhcpcd
```

```
subnet 192.168.100.0 netmask 255.255.255.0 {  
    ...  
    host pc221 {  
        hardware ethernet 00:08:83:2f:65:c0;  
        fixed-address 192.168.100.221;  
    }  
    host pc222 {  
        hardware ethernet 00:02:3f:7e:0f:45;  
        fixed-address 192.168.100.222;  
    }  
}
```

è possibile assegnare
un IP fisso sulla base
dell'indirizzo MAC

dhcpcd.conf – configurazione subnet multiple

Esempio di file di configurazione per dhcpcd

```
shared-network my_network {  
    subnet 192.168.100.0 netmask 255.255.255.0 {  
        ...  
    }  
    subnet 192.168.101.0 netmask 255.255.255.0 {  
        ...  
    }  
}
```

necessario se si vogliono definire subnet diverse sulla stessa rete fisica

(il nome my_network è simbolico)

Linux come client DHCP (CLI) – Avvio

dhclient -v <interface> da eseguire come root

```
[root@deis77 root]# dhclient -v eth1
Internet Software Consortium DHCP Client V3.0p11
...

Listening on LPF/eth1/00:a0:c9:ac:ff:a6
Sending on LPF/eth1/00:a0:c9:ac:ff:a6
Sending on Socket/fallback

DHCPDISCOVER on eth1 to 255.255.255.255 port 67 interval 6
DHCPOFFER from 192.168.100.73
DHCPREQUEST on eth1 to 255.255.255.255 port 67
DHCPACK from 192.168.100.73
bound to 192.168.100.112 -- renewal in 15 seconds.
```

Linux come client DHCP (CLI) – Chiusura

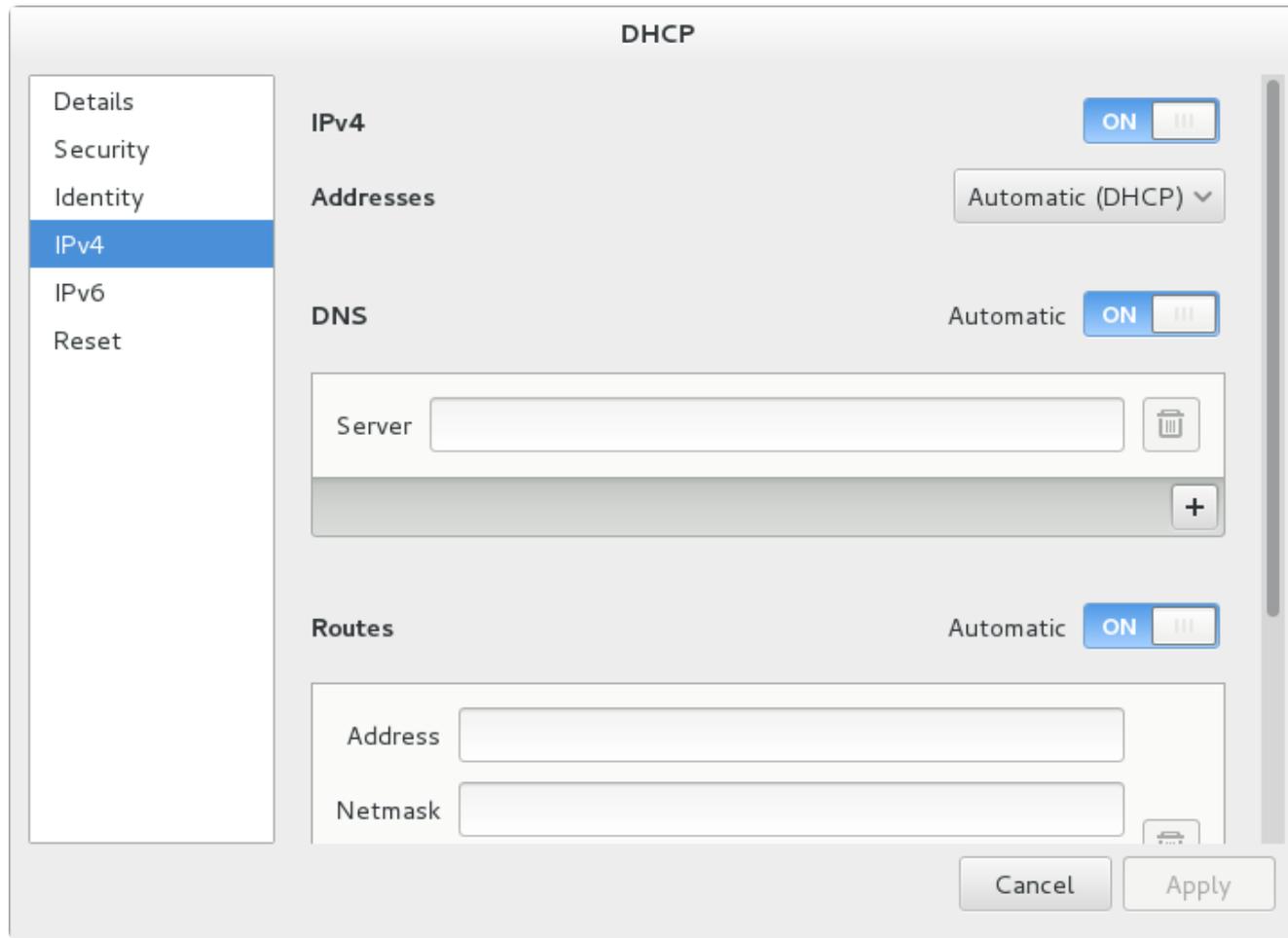
dhclient -v -r <interface>

```
[root@deis77 root]# dhclient -v -r eth1
Internet Software Consortium DHCP Client V3.0p11
Copyright 1995-2001 Internet Software Consortium.
All rights reserved.
For info, please visit http://www.isc.org/products/DHCP

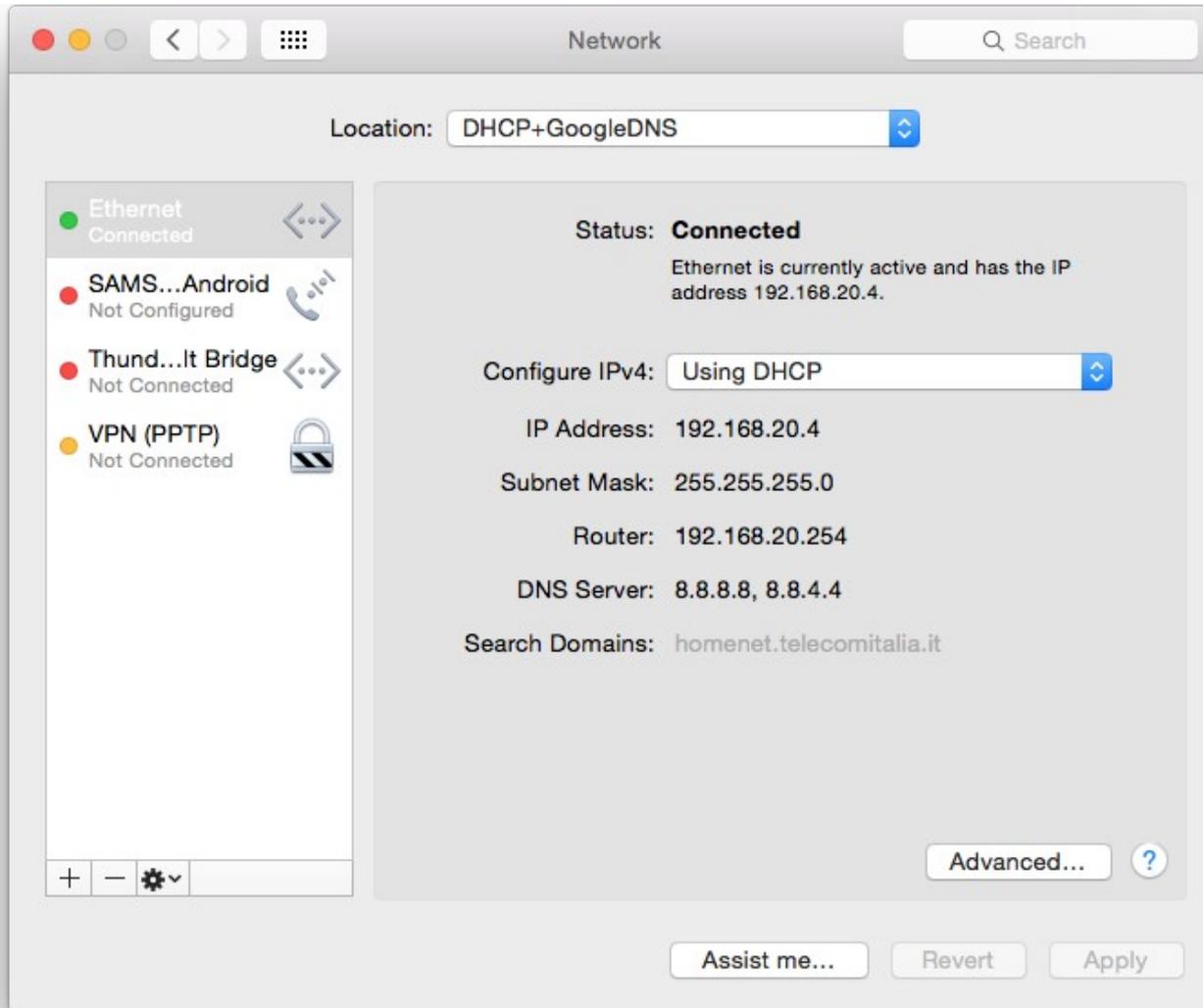
Listening on LPF/eth1/00:a0:c9:ac:ff:a6
Sending on    LPF/eth1/00:a0:c9:ac:ff:a6
Sending on    Socket/fallback
DHCPRELEASE on eth1 to 192.168.100.73 port 67
```

Registro degli assegnamenti in **/var/lib/dhclient/
dhclient.leases**

Linux come client DHCP (NM)



MAC come client DHCP



Windows come client DHCP

