

Avvertimento

Si consiglia di lavorare in un ambiente virtualizzato per evitare di danneggiare il proprio ambiente di lavoro o introdurre problemi di sicurezza.

Si informa che, in generale, le modifiche apportate col comando `ip` sono temporanee, e vengono pertanto perse al riavvio della macchina.

Riferimenti:

- <https://manpages.ubuntu.com/manpages/noble/man8/ip-link.8.html>
- <https://manpages.ubuntu.com/manpages/noble/man8/ip-address.8.html>
- <https://manpages.ubuntu.com/manpages/noble/man8/ip-route.8.html>
- <https://manpages.ubuntu.com/manpages/noble/man8/ip-neighbour.8.html>

Note:

- Nel comando `ip`, i prefissi possono essere abbreviati eliminando i byte 0 meno significativi.

Comandi utili

`ip link`

Il comando `ip link` ci permette di gestire le interfacce di rete attraverso svariati *sottocomandi*.

`ip address`

Il comando `ip address` ci permette di gestire i *protocol address* (ovvero gli indirizzi di rete; si veda la terminologia usata dal protocollo ARP) attraverso svariati *sottocomandi*.

`ip route`

Il comando `ip route` ci permette di gestire la tabella di instradamento attraverso vari sottocomandi

`ip neighbour`

Il comando `ip neigh` ci permette di gestire le tabelle ARP attraverso vari sottocomandi

Attività

Elencare le interfacce di rete

Il comando `ip link` senza ulteriori argomenti equivale a `ip link show` ed elenca pertanto tutte le interfacce di rete.

```
$ ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
   mode DEFAULT group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP
   mode DEFAULT group default qlen 1000
   link/ether 00:15:5d:f0:f1:dd brd ff:ff:ff:ff:ff:ff
```

interfaccia

Interfaccia di loopback (comunicazione locale all'host)

Interfaccia amministrativamente attiva

Livello fisico attivo (es. cavo collegato)

MTU

disciplina di coda: nessuna coda

Tipo di link (Ethernet), indirizzo MAC e indirizzo di broadcast

lunghezza della coda di pacchetti in attesa di trasmissione

supporta code multiple

collegamento operativo: es. una interfaccia wifi può essere attiva ma non operativa, per esempio perché non è associata ad alcun access point

Supporta pacchetti broadcast e multicast

Si può aggiungere l'opzione `-s` per visualizzare delle statistiche.

```
$ ip -s link show dev eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP
   mode DEFAULT group default qlen 1000
   link/ether 00:15:5d:f0:f6:cc brd ff:ff:ff:ff:ff:ff
RX:  bytes packets errors dropped missed mcst
    516554    1669      0      0      0     513
TX:  bytes packets errors dropped carrier collsns
    97370    1173      0      0      0      0
```

Frame ricevuti errati (es. CRC errato)

Pacchetti multicast ricevuti

Frame non elaborati in tempo dall'host

Numero di errori di trasmissione per perdita portante

ricezione

trasmissione

byte (escluso campo CRC)

Problemi di trasmissione

Frame ricevuti ma non elaborati

Pacchetti scartati prima di essere trasmessi

Frame corretti ricevuti (anche se dopo scartati)

Frame passati con successo all'adattatore di rete

Collisions

Riferimento per le statistiche: <https://docs.kernel.org/networking/statistics.html#struct-rtnl-link-stats64>

Possibilità di filtrare (i diversi filtri possono essere *congiunti* mettendoli uno dopo l'altro):

- Solo interfacce attive: `ip link show up`
- Solo una certa interfaccia: `ip link show dev eth0`

Nota: abbiamo trovato (in rosso) l'indirizzo MAC associato all'interfaccia di rete `eth0`.

Disattivare interfaccia di rete

Disattivare un'interfaccia di rete causa la rimozione di tutte le rotte che la attraversano. Conviene pertanto salvare la tabella di instradamento su file e poi ripristinarla.

```
$ ip route save > route.txt
```

Prima di ripristinare la tabella di instradamento, può essere necessario svuotarla per evitare errori dovuti a rotte già presenti (es. quelle aggiunte quando si attiva un'interfaccia, relative alle sottoreti direttamente connesse).

```
$ sudo ip route flush table main
```

Ripristina la tabella di instradamento.

```
$ sudo ip route restore < route.txt
```

Disattiva l'interfaccia `eth0`.

```
$ sudo ip link set eth0 down
```

Per riattivare la interfaccia:

```
$ sudo ip link set eth0 up
```

Prima di ripristinare la tabella di instradamento, può essere necessario svuotarla per evitare errori dovuti a rotte già presenti (es. quelle aggiunte quando si attiva un'interfaccia, relative alle sottoreti direttamente connesse).

```
$ sudo ip route flush table main
```

Ripristina la tabella di instradamento.

```
$ sudo ip route restore < route.txt
```

Elencare gli indirizzi

Il comando `ip address` (spesso abbreviato in `ip addr`) senza ulteriori argomenti equivale a `ip address show` ed elenca pertanto gli indirizzi associati a tutte le interfacce.

```
$ ip address
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet 10.255.255.254/32 brd 10.255.255.254 scope global lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP
group default qlen 1000
    link/ether 00:15:5d:f0:f1:dd brd ff:ff:ff:ff:ff:ff
    inet 172.21.242.109/20 brd 172.21.255.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::215:5dff:fef0:f1dd/64 scope link
        valid_lft forever preferred_lft forever
```

Indirizzo IP (inet) 172.21.242.109 con maschera
di rete /20, indirizzo di broadcast
172.21.255.255 e scope globale.

periodo di validità dopo il
quale viene cancellato

Periodo dopo il
quale non viene
più usato

Nota: abbiamo trovato (in rosso) l'indirizzo IP associato all'interfaccia di rete `eth0`.

Il campo `scope` l'ambito in cui è valido questo indirizzo. Gli ambiti disponibili sono elencati nel file `/etc/iproute2/rt_scopes`. I valori predefiniti dell'ambito sono:

- `global` - l'indirizzo è valido a livello globale.
- `site` - (solo IPv6, deprecato) l'indirizzo è locale al sito, cioè è valido all'interno del sito.
- `link` - l'indirizzo è locale al link, cioè è valido solo su questo dispositivo di rete.
- `host` - l'indirizzo è valido solo all'interno di questo host.

Possibilità di filtrare (i diversi filtri possono essere *congiunti* mettendoli uno dopo l'altro):

- Solo interfacce attive: `ip address show up`
- Solo una certa interfaccia: `ip address show dev eth0`
- Solo un certo scope: `ip address show scope global`
- Solo un certo prefisso: `ip address show to 172.0.0.0/8`

Mostrare la tabella di instradamento

Il comando `ip route` senza ulteriori argomenti equivale a `ip route show` ed elenca pertanto tutte le rotte nella tabella di instradamento principale (main).

`$ ip route`

default via **172.21.240.1** dev eth0 proto kernel

172.21.240.0/20 dev eth0 proto kernel scope link src 172.21.242.109

Nota: abbiamo trovato (in rosso) l'indirizzo IP del router di default.

Possibilità di filtrare (i diversi filtri possono essere *congiunti* mettendoli uno dopo l'altro):

- Rotte che attraversano una certa interfaccia: `ip route show dev eth0`
- Rotte il cui next hop è selezionato da un prefisso (se si omette `/n` si intende `/32`, cioè indirizzo IP esatto): `ip route show via 172.0.0.0/8`
- Rotte per un dato insieme di destinazioni: negli esempi viene passato un prefisso `/n` come argomento, ma si può usare anche un indirizzo IP interpretato come `/32`.

Negli esempi sottostanti assumiamo questa tabella di instradamento:

```
default via 172.21.240.1 dev eth0 proto kernel
172.21.240.0/20 dev eth0 proto kernel scope link src 172.21.242.109
```

- rotte il cui prefisso è un'estensione di quello passato come argomento:

```
$ ip route show to root 0/0
```

```
default via 172.21.240.1 dev eth0 proto kernel
```

```
172.21.240.0/20 dev eth0 proto kernel scope link src 172.21.242.109
```

Altro esempio:

```
$ ip route show to root 172.0.0.0/8
172.21.240.0/20 dev eth0 proto kernel scope link src 172.21.242.109
```

- rotte per un determinato prefisso: va usato *exact*, ma viene assunto come default se non c'è altro modificatore:

```
$ ip route show to exact 0.0.0.0/0
default via 172.21.240.1 dev eth0 proto kernel
```

oppure

```
$ ip route show to 0.0.0.0/0
default via 172.21.240.1 dev eth0 proto kernel
```

oppure

```
$ ip route show 0.0.0.0/0
default via 172.21.240.1 dev eth0 proto kernel
```

- rotte associate a un prefisso del prefisso passato:

```
$ ip route show to match 172.0.0.0/8
default via 172.21.240.1 dev eth0 proto kernel
```

- Rotte il cui campo src è selezionato da un prefisso (se si omette /n si intende /32, cioè indirizzo IP esatto): `ip route show src 172.0.0.0/8`

Ottenere una singola rotta per una destinazione

Il comando `ip route get` permette di ottenere la rotta per una data destinazione, come vista dal kernel dopo aver risolto le informazioni contenute nella tabella di instradamento.

È quella usata per l'inoltro verso quella destinazione.

Tiene conto del *longest prefix match*.

```
ip route get 8.8.8.8
8.8.8.8 via 172.21.240.1 dev eth0 src 172.21.242.109 uid 1000
cache
```

Inviare un ping a 8.8.8.8

Si faccia riferimento ai dati appuntati nelle attività precedenti.

```
$ ping -c 1 8.8.8.8
```

Verificare con wireshark quali sono gli indirizzi a tutti i livelli. Cosa indica l'indirizzo MAC di destinazione?

Mostrare la tabella ARP

Il comando `ip neigh` senza alcun parametro è equivalente a `ip neigh show` che mostra il contenuto di tutte le tabelle ARP.

```
$ ip neigh
172.21.240.1 dev eth0 lladdr 00:15:5d:10:bd:f1 REACHABLE
```

La riga per un certo indirizzo IP ci fornisce l'interfaccia (dev), l'indirizzo MAC (lladdr) e lo stato (in questo caso REACHABLE).

Si dovrebbe vedere che il mac trovato nell'attività precedente è quello del router di default.

È possibile filtrare per una combinazione di:

- interfaccia: `ip neigh show dev eth0`
- selettore sull'indirizzo IP: `ip neigh show to 172.0.0.0/8`
- stato: `ip neigh show nud reachable`

Lookup nella tabella ARP

```
$ ip neigh get 172.21.240.1 dev eth0
172.21.240.1 dev eth0 lladdr 00:15:5d:10:f5:5b REACHABLE
```

Nota: non usa il protocollo ARP per creare una nuova entry

Cancellare tutte le voci (non permanenti) della tabella ARP

Nota: il comando `flush` ha gli stessi argomenti di `show`.

```
sudo ip neigh flush dev eth0
```

Eventualmente possiamo aggiungere un filtro sullo stato,

```
sudo ip neigh flush nud stale dev eth0
```