

LinuxDay 2002 Ferrara

iproute2: un unico tool per il networking

Ing. Simone Piunno

La pila ISO/OSI

7. applicazione (es. API BSD)

6. presentazione (es. HTML)

5. sessione (es. HTTP)

4. trasporto (es. TCP)

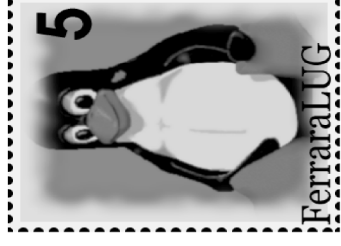
3. rete (es. IP)

2. collegamento (es. 802.3)

1. fisico (es. UTP cat-5)

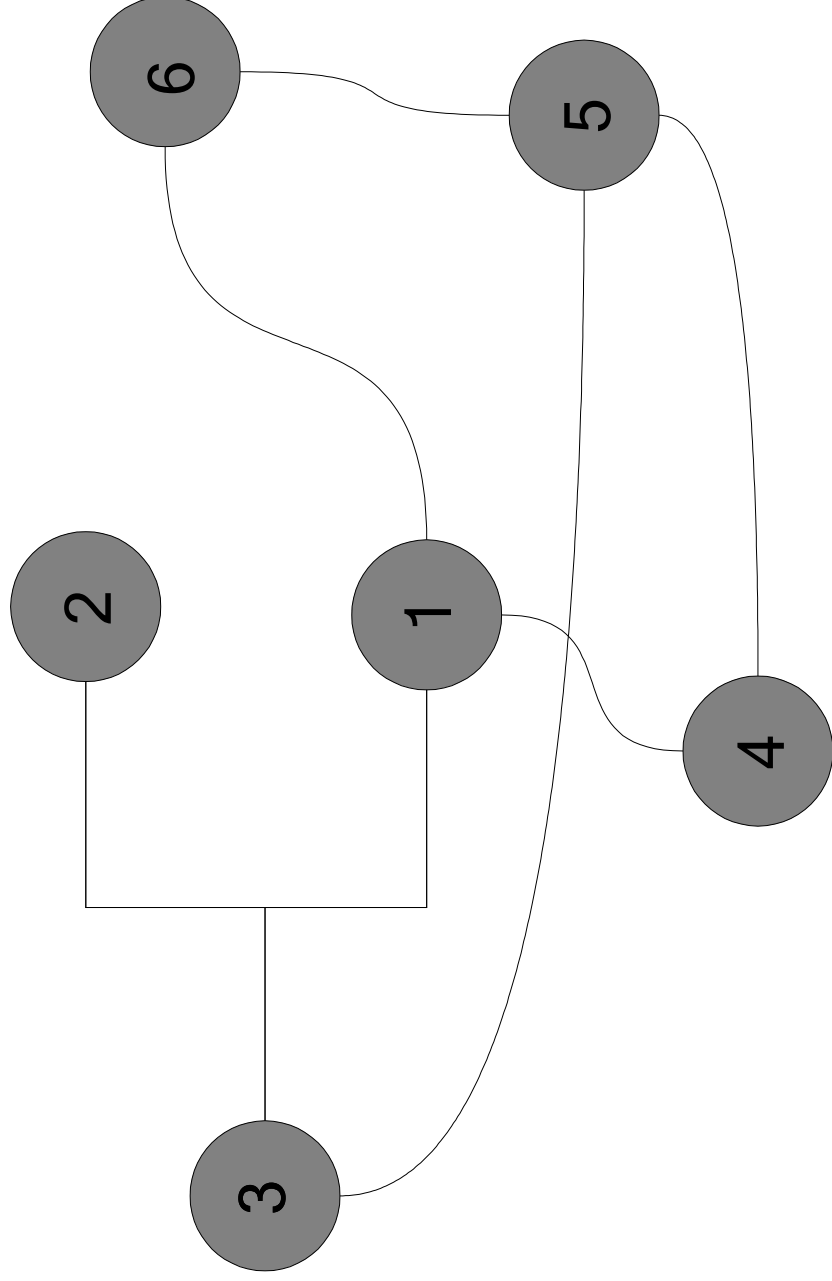
Incapsulamento

Mitt: 192.168.0.1



Per il Sig.
192.168.5.23

La rete è un grafo



I vecchi net-tools

- **ifconfig**
 - Abilitazione interfacce e parametri L2
 - Assegnazione indirizzi IP
- **route**
 - Come si raggiunge una destinazione IP
- **arp**
 - Come interagisce IP con L2
- **iptunnel**
 - Tunnel IP over IP
- **ipmaddr**
 - Assegnazione indirizzi multicast

Il nuovo tool

- iproute2 racchiude le funzioni di tutti gli altri vecchi tool
- usa RT-NetLink Socket
- Sviluppato da Alexey Kuznetsov a partire dal kernel 2.4

Attivazione / ifconfig

- Con ifconfig:

```
# ifconfig eth0 192.168.0.1 netmask 255.255.255.0 up
# ifconfig eth0
eth0  Link encap:Ethernet  HWaddr 00:E0:18:19:02:14
      inet addr:192.168.0.1  Bcast:192.168.0.255  Mask:255.255.255.0
      UP BROADCAST NOTRAILERS RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:682 errors:4 dropped:8 overruns:4 frame:0
      TX packets:197 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:100
      RX bytes:315304 (307.9 Kb)  TX bytes:17928 (17.5 Kb)
      Interrupt:9 Base address:0x2c00
```

Attivazione/ip

- Con ip link e ip addr:

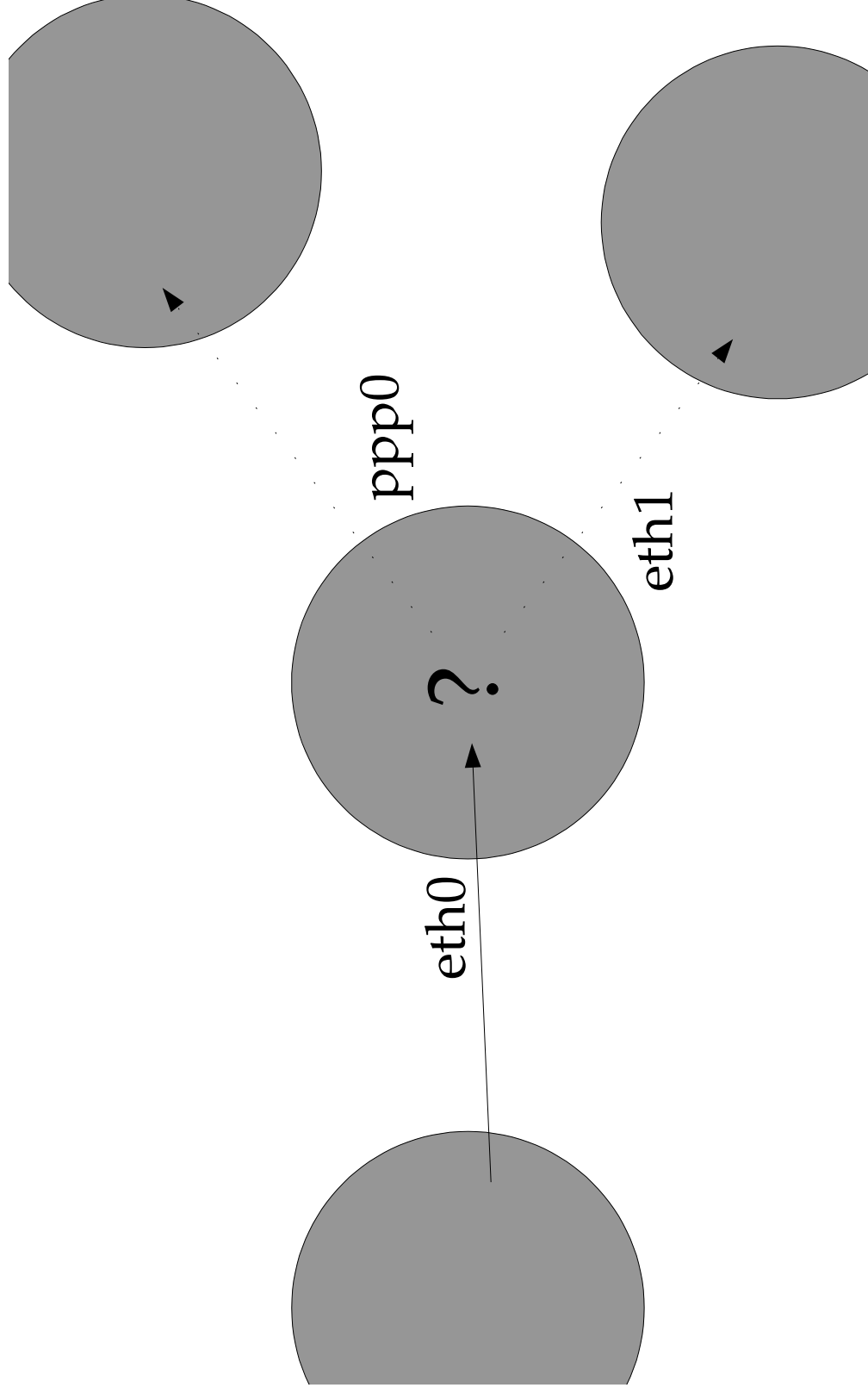
```
# ip addr add 192.168.0.1/24 dev eth0
# ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,NOTRAILERS,UP> mtu 1500 qdisc
  pfifo_fast qlen 100
  link/ether 00:e0:18:19:02:14 brd ff:ff:ff:ff:ff:ff
  inet 192.168.9.9/24 brd 192.168.9.255 scope global eth0
```

```
# ip addr link set eth0 up
# ip -s link show eth0
2: eth0: <BROADCAST,MULTICAST,NOTRAILERS,UP> mtu 1500 qdisc
  pfifo_fast qlen 100
  link/ether 00:e0:18:19:02:14 brd ff:ff:ff:ff:ff:ff
  RX: bytes  packets  errors  dropped overrun mcast
  316159    687      4      0      0      0
  TX: bytes  packets  errors  dropped carrier collsns
  17928     197      0      0      0      0
```


Multipli indirizzi IP

- con ifconfig:
 - Un indirizzo per interfaccia
 - Trucco delle interfacce alias (es. eth0:1)
- con ip:
 - Multipli indirizzi per interfaccia, anche misti IPv4 e IPv6
 - Etichetta arbitraria assegnabile ad ogni indirizzo

Dirigere il traffico



Dirigere il traffico

- **Con route:**

```
# route add -net 192.168.1.0 gw 192.168.0.8 nm 255.255.255.0 dev eth0
# route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
192.168.1.0 192.168.0.8 255.255.255.0 UG 0 0 0 eth0
192.168.0.0 * 255.255.255.0 U 0 0 0 eth0
127.0.0.0 * 255.0.0.0 U 0 0 0 lo
```

- **Con ip route:**

```
# ip route add 192.168.1.0/24 via 192.168.0.8
# ip route show
192.168.1.0/24 via 192.168.0.8 dev eth0
192.168.0.0/24 dev eth0 proto kernel scope link src 192.168.0.1
127.0.0.0/8 dev lo scope link
```

Routing multi-tabella

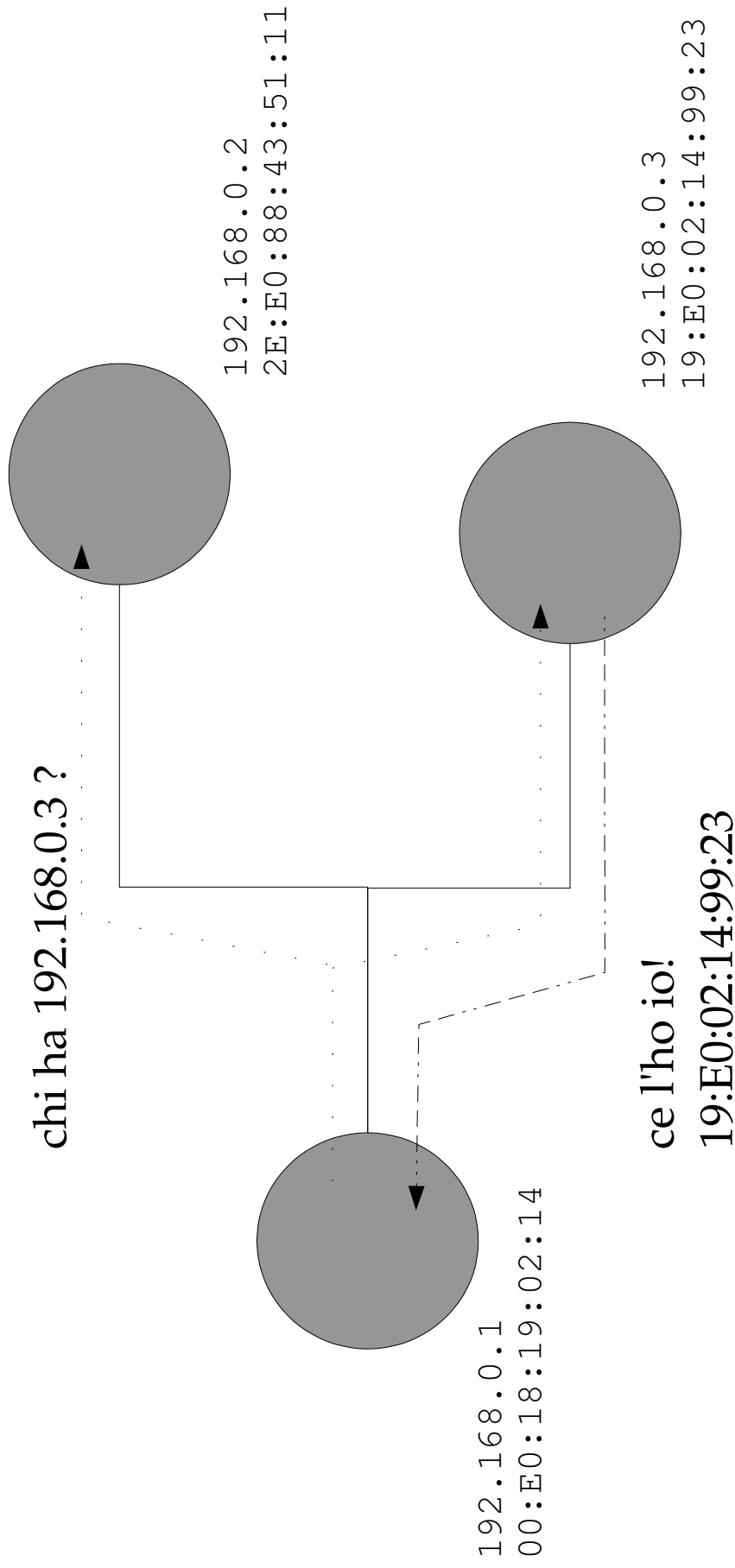
ip route permette tabelle multiple:

```
# ip route show table local
broadcast 192.168.0.255 dev eth0 proto kernel scope link src 192.168.0.1
broadcast 127.255.255.255 dev lo proto kernel scope link src 127.0.0.1
broadcast 192.168.0.0 dev eth0 proto kernel scope link src 192.168.0.1
local 192.168.0.1 dev eth0 proto kernel scope host src 192.168.0.1
broadcast 127.0.0.0 dev lo proto kernel scope link src 127.0.0.1
local 127.0.0.1 dev lo proto kernel scope host src 127.0.0.1
local 127.0.0.0/8 dev lo proto kernel scope host src 127.0.0.1

# ip route show table main
192.168.1.0/24 via 192.168.0.8 dev eth0
192.168.0.0/24 dev eth0 proto kernel scope link src 192.168.0.1
127.0.0.0/8 dev lo scope link

# ip rule add from 195.194.193.0/24 table 240
# ip rule list
0:      from all lookup local
32765:  from 195.194.193.0/24 lookup 240
32766:  from all lookup main
32767:  from all lookup 253
```

Address Resolution L2



Address Resolution L2

con arp:

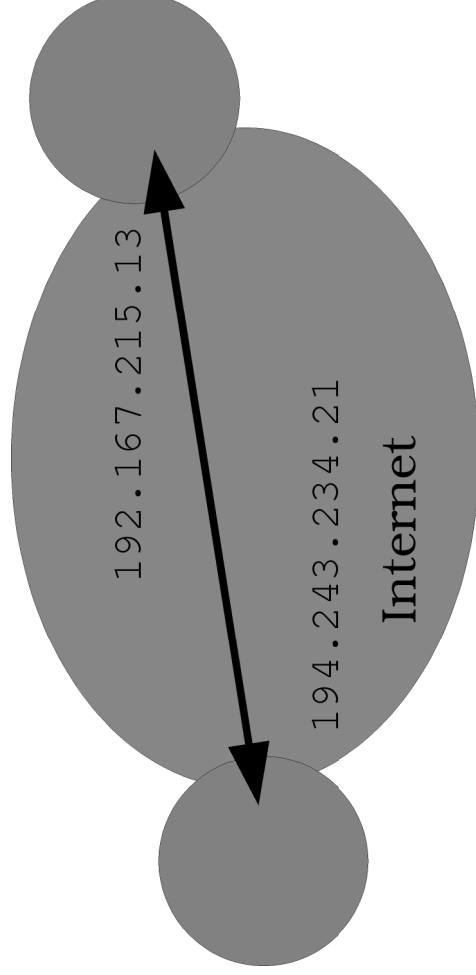
```
# arp -n
Address          HWtype  HWaddress      Flags Mask    Iface
192.168.0.8      ether   00:48:54:1B:25:30  C          eth0
192.168.0.31     ether   00:2c:74:33:37:39  C          eth0
```

con ip neigh:

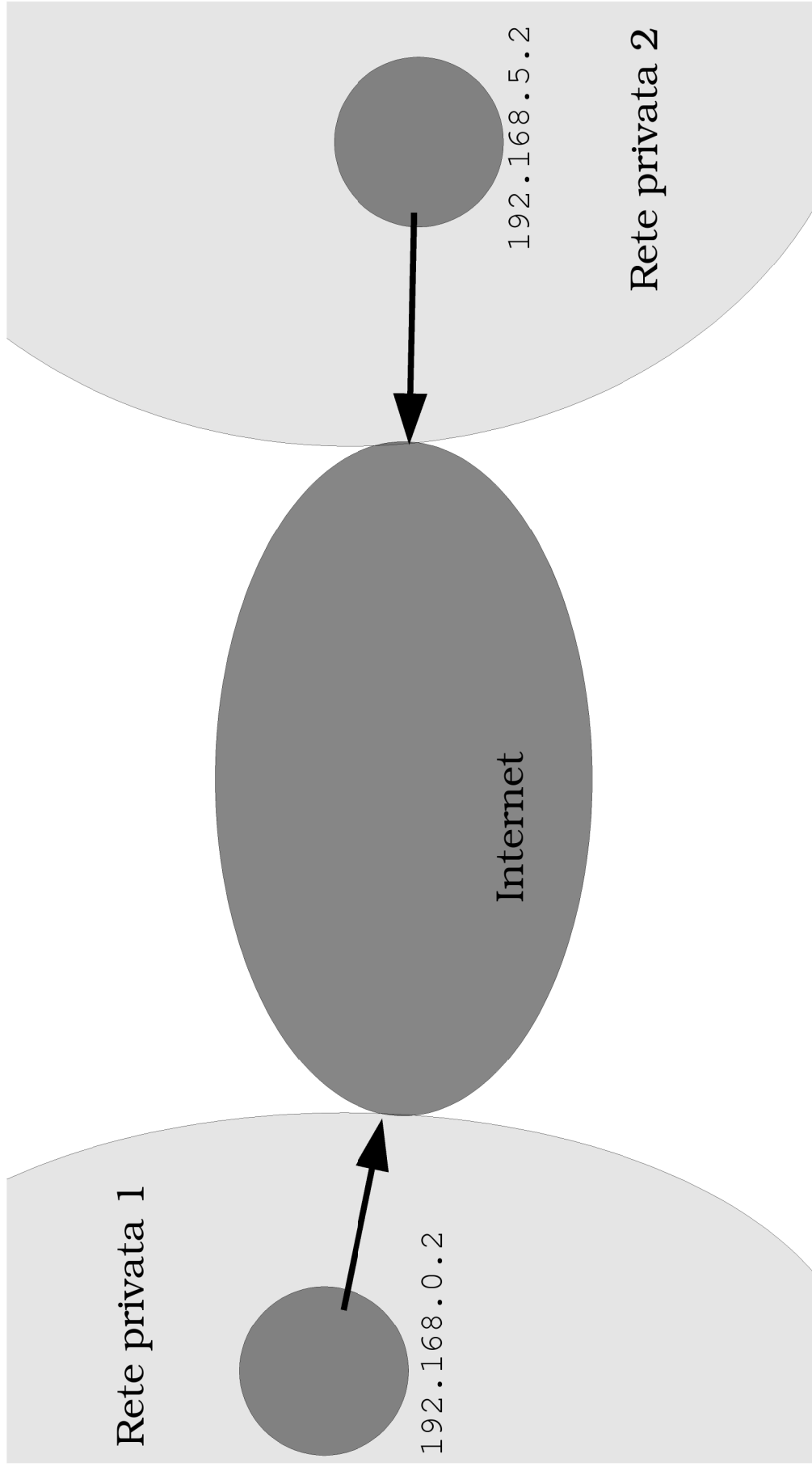
```
# ip neigh show
192.168.0.8 dev eth0 lladdr 00:48:54:1b:25:30 nud stale
192.168.0.31 dev eth0 lladdr 00:2c:74:33:37:39 nud stale

# ip neigh add proxy 192.168.2.5 dev eth0
```

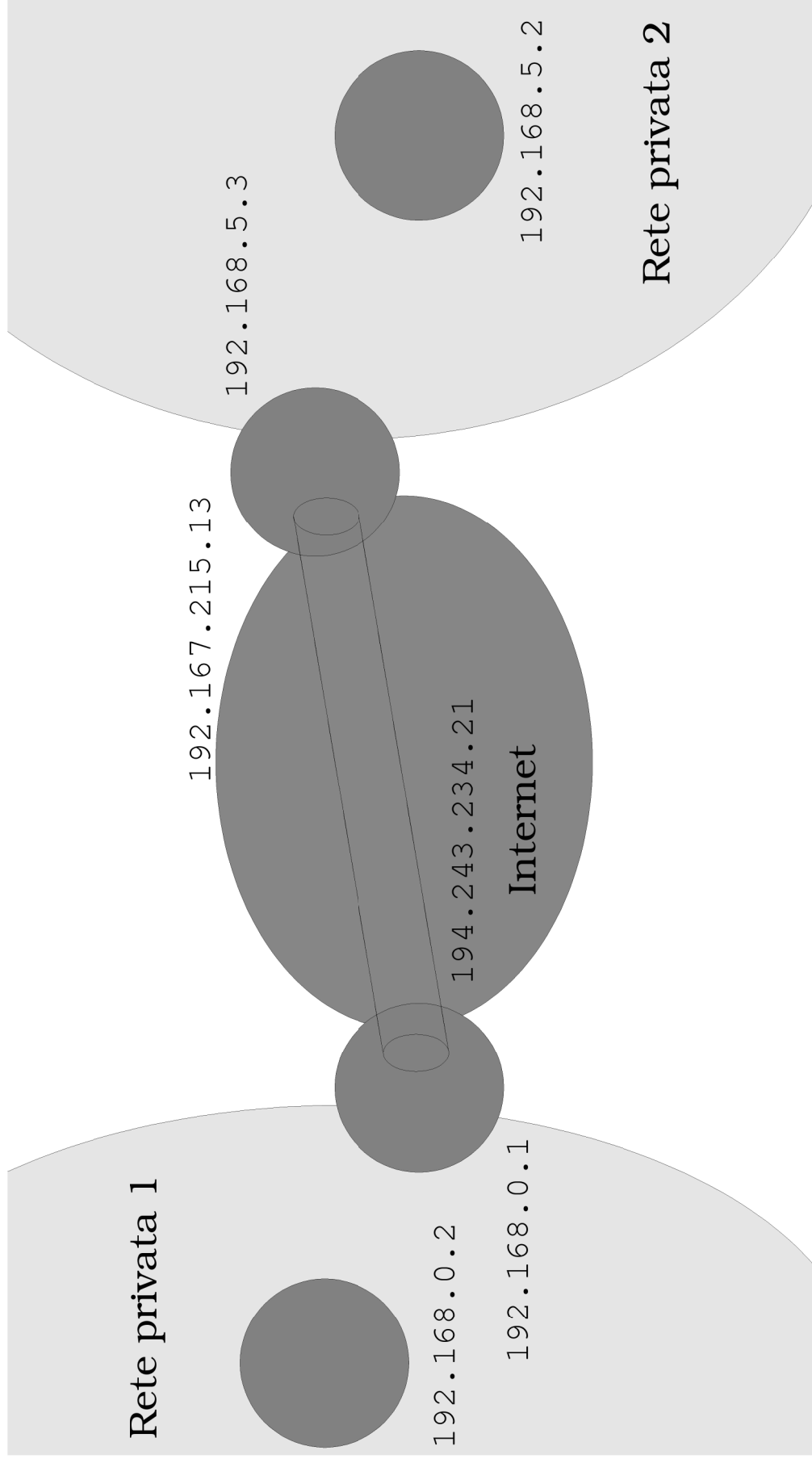
Tunneling/1



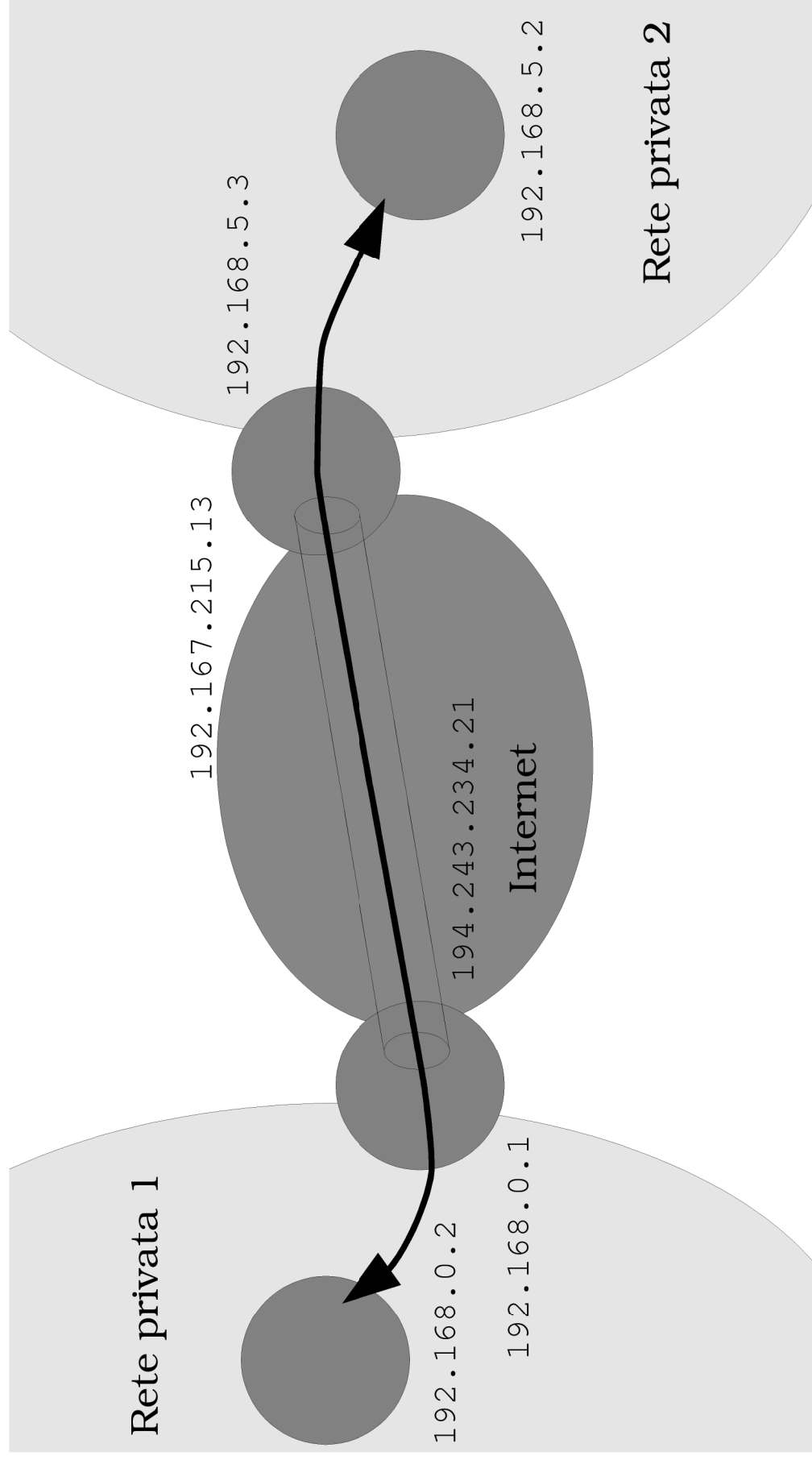
Tunneling/2



Tunneling/3



Tunneling/4



Tunneling / stack protocollare

L4 (es. TCP)
IPv4 o IPv6
IPv4 (o IPv6)
L2 (es. PPP)

Tunnel
IPIP

L4 (es. TCP)
any
GRE
IPv4
L2 (es. PPP)

Tunnel
GRE

Tunneling / configurazione

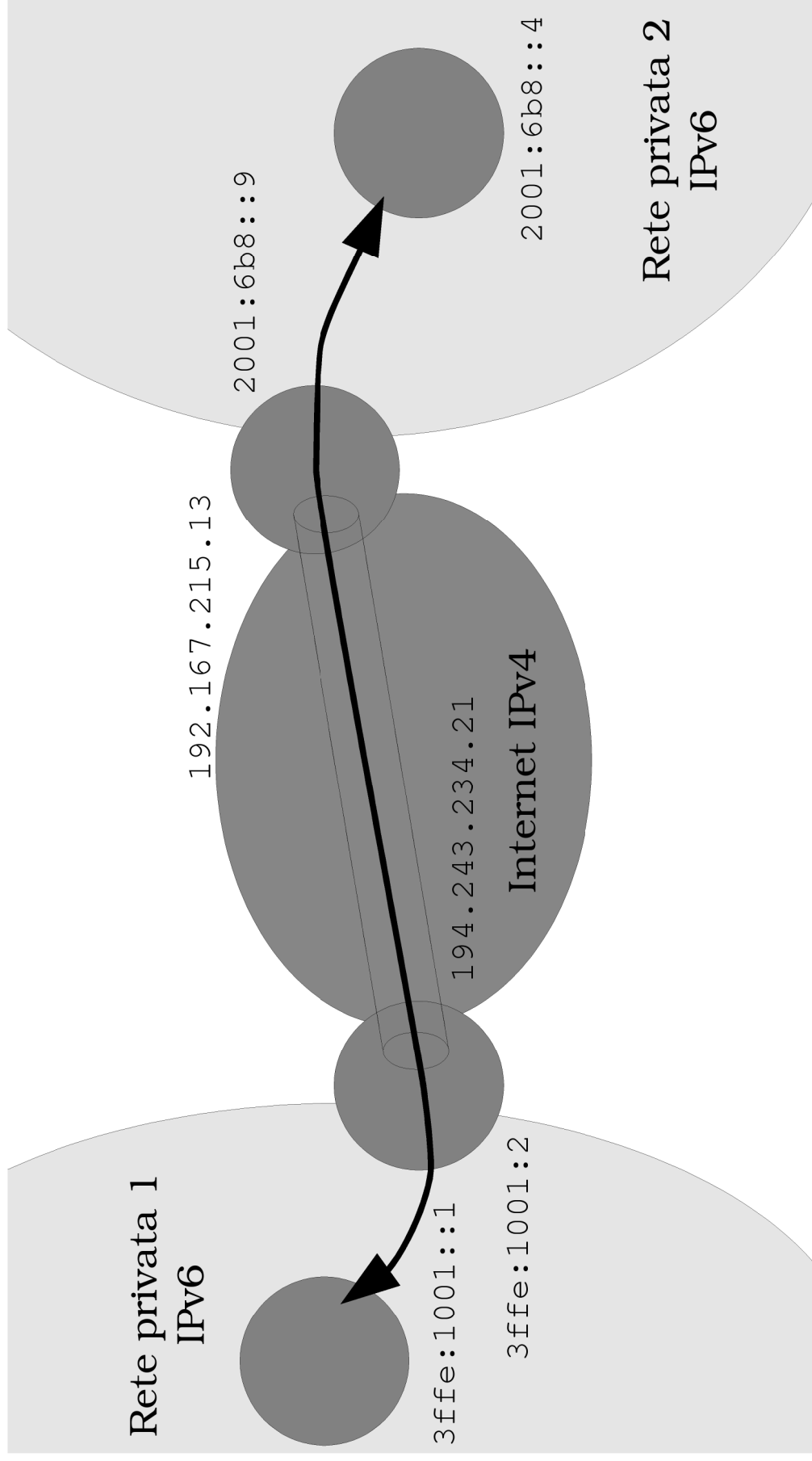
- con iptunnel:

```
# iptunnel add pippo mode gre remote 194.243.234.21 ttl 64
# iptunnel show pippo
pippo: gre/ip remote 194.243.234.21 local any dev eth0 ttl 64
```

- con ip tunnel:

```
# ip tunnel add pippo mode gre remote 194.243.234.21 ttl 64
# ip tunnel show pippo
pippo: gre/ip remote 194.243.234.21 local any dev eth0 ttl 64
```

Tunneling / IPv6



Conclusion

- iproute2 riunisce la gestione del networking (L3) in una interfaccia uniforme
- Vengono introdotti ulteriori gradi di libertà rispetto ai tool precedenti
- Risulta molto comodo per la transizione a IPv6