Protocollo ARP
IP forwarding

A.A. 2018/2019

Walter Cerroni

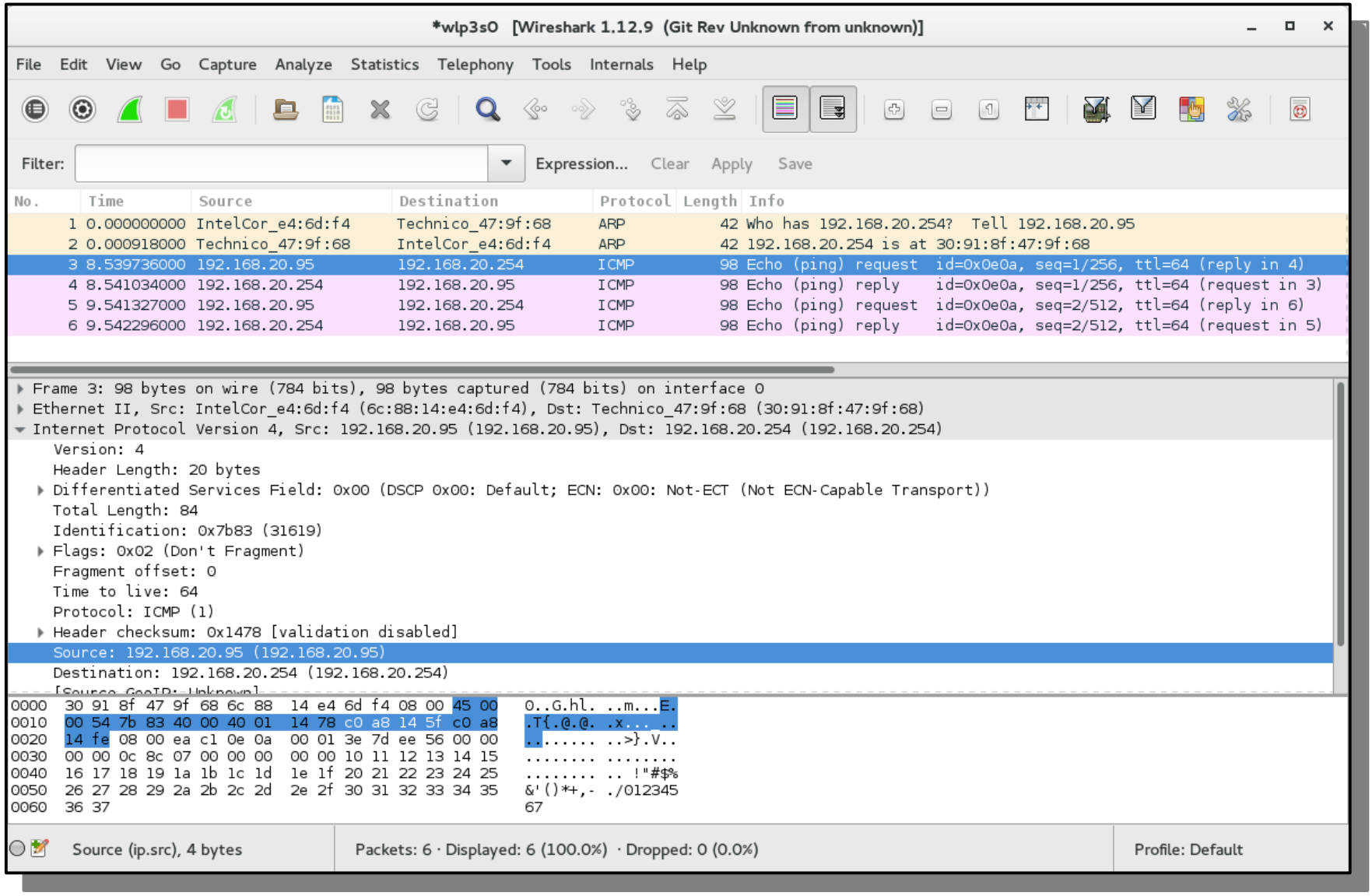
Reti IP come insiemi di indirizzi

- Una rete IP è individuata dall'**indirizzo di rete**, che è quello con i bit dell'Host-ID tutti a zero (in genere non si assegna ad un'interfaccia):
  - 192.168.8.0/24
  - 10.0.0.4/30
- Tutti gli indirizzi che hanno lo stesso Network-ID appartengono alla stessa rete IP
  - comunicano tra loro tramite consegna diretta dei pacchetti
  - comunicano con gli altri tramite consegna indiretta attraverso un gateway
- L'**indirizzo di broadcast**, che serve per raggiungere tutti gli host di una rete IP, è quello con i bit dell'Host-ID tutti a uno:
  - 192.168.8.255/24
  - 10.0.0.7/30
- In genere l'indirizzo immediatamente prima del broadcast è assegnato al **gateway di default** usato per le indirect delivery
  - 192.168.8.254/24
  - 10.0.0.6/30 (anche se in un collegamento punto-punto ha meno senso)

Indirizzi e interfacce di rete

- L’indirizzo IP specifica sia la rete sia l’host
- In realtà si riferisce ad una delle interfacce di rete dell’host
- Non identifica un host individuale, ma una interfaccia di rete
- **Multi-homed host**: host con due o più interfacce di rete → usa più indirizzi IP
- Un router che collega N reti ha almeno N distinti indirizzi IP, uno per ogni interfaccia di rete
- Le interfacce hanno anche indirizzi fisici, in base al protocollo di strato 2 adottato

Indirizzi visibili da Wireshark



Indirizzi visibili da tcpdump

root@deisnet99:~

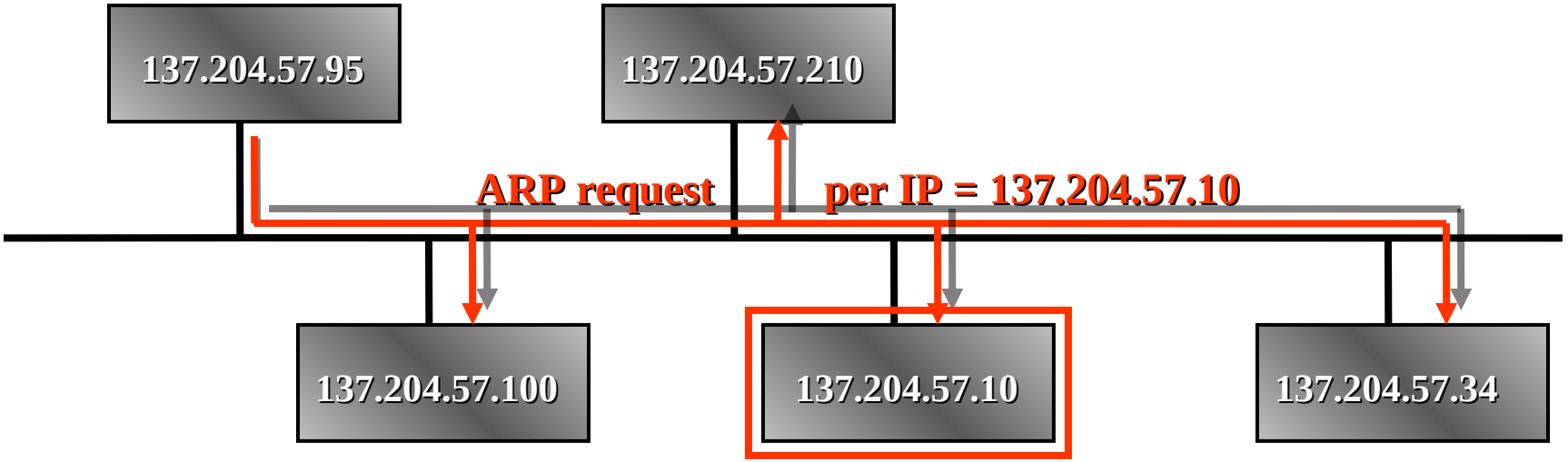
File Edit View Search Terminal Help

```
[root@deisnet99 ~]# tcpdump -nnev -i wlp3s0
tcpdump: listening on wlp3s0, link-type EN10MB (Ethernet), capture size 262144 bytes
11:40:43.017992 6c:88:14:e4:6d:f4 > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 42: Ethernet (l
en 6), IPv4 (len 4), Request who-has 192.168.20.254 tell 192.168.20.95, length 28
11:40:43.021398 30:91:8f:47:9f:68 > 6c:88:14:e4:6d:f4, ethertype ARP (0x0806), length 42: Ethernet (l
en 6), IPv4 (len 4), Reply 192.168.20.254 is-at 30:91:8f:47:9f:68, length 28
11:40:43.021438 6c:88:14:e4:6d:f4 > 30:91:8f:47:9f:68, ethertype IPv4 (0x0800), length 98: (tos 0x0,
ttl 64, id 16098, offset 0, flags [DF], proto ICMP (1), length 84)
    192.168.20.95 > 192.168.20.254: ICMP echo request, id 3714, seq 1, length 64
11:40:43.023198 30:91:8f:47:9f:68 > 6c:88:14:e4:6d:f4, ethertype IPv4 (0x0800), length 98: (tos 0xa0,
 ttl 64, id 21309, offset 0, flags [none], proto ICMP (1), length 84)
    192.168.20.254 > 192.168.20.95: ICMP echo reply, id 3714, seq 1, length 64
11:40:44.019479 6c:88:14:e4:6d:f4 > 30:91:8f:47:9f:68, ethertype IPv4 (0x0800), length 98: (tos 0x0,
ttl 64, id 16150, offset 0, flags [DF], proto ICMP (1), length 84)
    192.168.20.95 > 192.168.20.254: ICMP echo request, id 3714, seq 2, length 64
11:40:44.020483 30:91:8f:47:9f:68 > 6c:88:14:e4:6d:f4, ethertype IPv4 (0x0800), length 98: (tos 0xa0,
 ttl 64, id 21310, offset 0, flags [none], proto ICMP (1), length 84)
    192.168.20.254 > 192.168.20.95: ICMP echo reply, id 3714, seq 2, length 64
11:40:45.020762 6c:88:14:e4:6d:f4 > 30:91:8f:47:9f:68, ethertype IPv4 (0x0800), length 98: (tos 0x0,
ttl 64, id 16217, offset 0, flags [DF], proto ICMP (1), length 84)
    192.168.20.95 > 192.168.20.254: ICMP echo request, id 3714, seq 3, length 64
11:40:45.021745 30:91:8f:47:9f:68 > 6c:88:14:e4:6d:f4, ethertype IPv4 (0x0800), length 98: (tos 0xa0,
 ttl 64, id 21311, offset 0, flags [none], proto ICMP (1), length 84)
    192.168.20.254 > 192.168.20.95: ICMP echo reply, id 3714, seq 3, length 64
^C
8 packets captured
8 packets received by filter
0 packets dropped by kernel
[root@deisnet99 ~]#
[root@deisnet99 ~]#
```

Relazione Indirizzi Fisici – Indirizzi IP

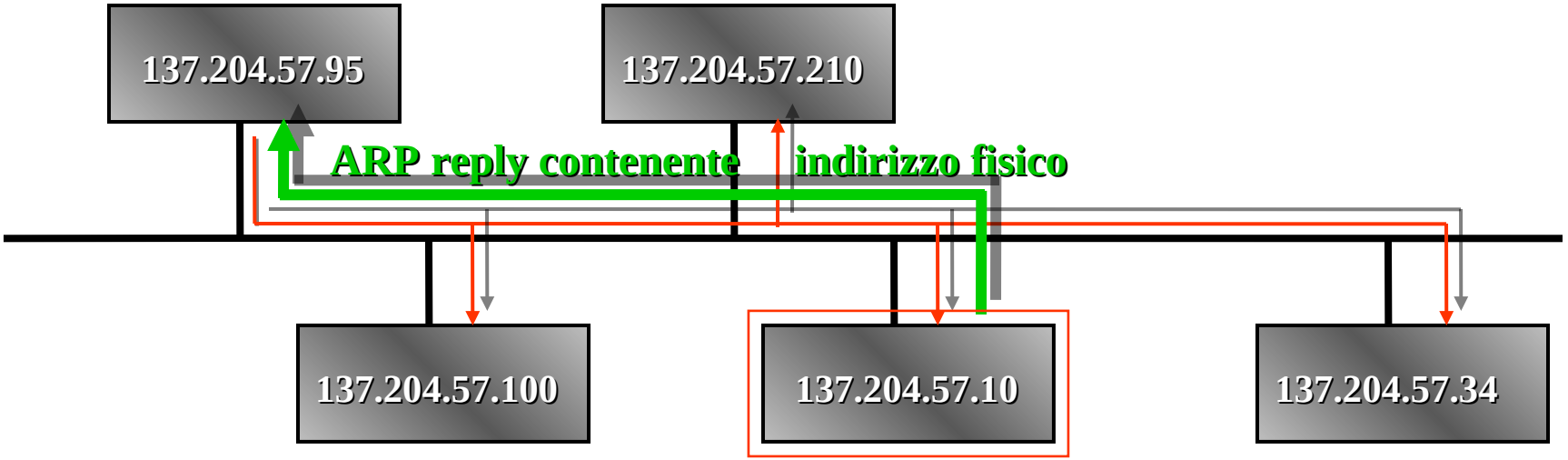
- Software di basso livello nasconde gli indirizzi fisici e consente ai livelli superiori di lavorare solo con indirizzi IP
- Gli host comunicano attraverso una **rete fisica** (ad es. LAN) quindi devono conoscere reciprocamente gli indirizzi fisici
- L'host A vuole mandare datagrammi a B, che si trova sulla stessa rete e di cui conosce solo l'indirizzo IP
- Come si ottiene l'indirizzo fisico di B dato il suo indirizzo IP?

Address Resolution Protocol – ARP (RFC 826)



- Il nodo sorgente invia una trama broadcast (**ARP request**) contenente l'indirizzo IP del nodo destinazione
- Tutte le stazioni della rete locale leggono la trama broadcast

Address Resolution Protocol – ARP



- Il destinatario risponde al mittente, inviando un messaggio (**ARP reply**) che contiene il proprio indirizzo fisico
- Con questo messaggio l'host sorgente è in grado di associare l'appropriato indirizzo fisico all'IP destinazione
- Ogni host mantiene una tabella (**cache ARP**) con le corrispondenze fra indirizzi logici e fisici

Il comando arp

- Il comando **arp -n** visualizza il contenuto della cache ARP con le corrispondenze tra indirizzi IP e MAC conosciuti e l'interfaccia su cui tale corrispondenza ha validità
- Il comando **ip neigh** svolge le stesse funzioni di **arp**

```
[labuser@netlab07 ~]$ arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.8.254            ether   00:02:B3:D7:EF:15   C                     eth1
192.168.8.3              ether   00:07:E9:89:AC:CF   C                     eth1
192.168.8.8                      (incomplete)                              eth1
192.168.8.5              ether   00:07:E9:E7:4A:23   C                     eth1
[labuser@netlab07 ~]$
```

```
[labuser@netlab07 ~]$ ip neigh show
192.168.8.254 dev eth1 lladdr 00:02:b3:d7:ef:15 REACHABLE
192.168.8.3 dev eth1 lladdr 00:07:e9:89:ac:cf REACHABLE
192.168.8.8 dev eth1  FAILED
192.168.8.5 dev eth1 lladdr 00:07:e9:e7:4a:23 REACHABLE
[labuser@netlab07 ~]$
```

I comandi arp e arping

- Per eliminare una voce dalla cache ARP:

  arp -i <interface> -d <IP_addr>

- Per aggiungere manualmente una voce alla cache ARP:

  arp -i <interface> -s <IP_addr> <HW_addr> [temp]
  (temporanea se si specifica il flag **temp**)

- L'opzione **-D <interface>** equivale a specificare l'indirizzo MAC **<HW_addr>** dell'interfaccia specificata

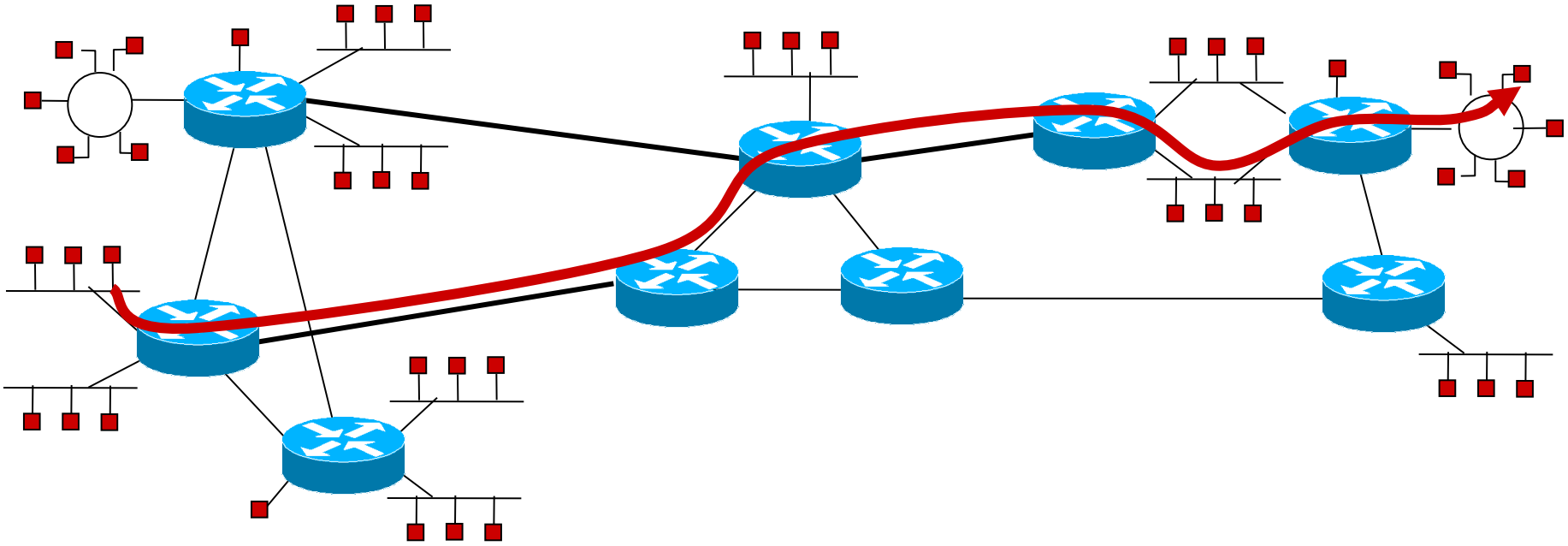
- Per aggiungere una voce di proxy-ARP:

  arp -i <interface> -s <IP_addr> -D <interface> pub
  <interface> indica l'interfaccia su cui rispondere alle richieste
  <IP_addr> indica l'indirizzo per conto del quale si fa proxy ARP
  (richiede l'attivazione dell'*IP forwarding*)

- Per generare delle ARP request:

  arping -I <interface> <IP_addr>

IP: instradamento dei datagrammi

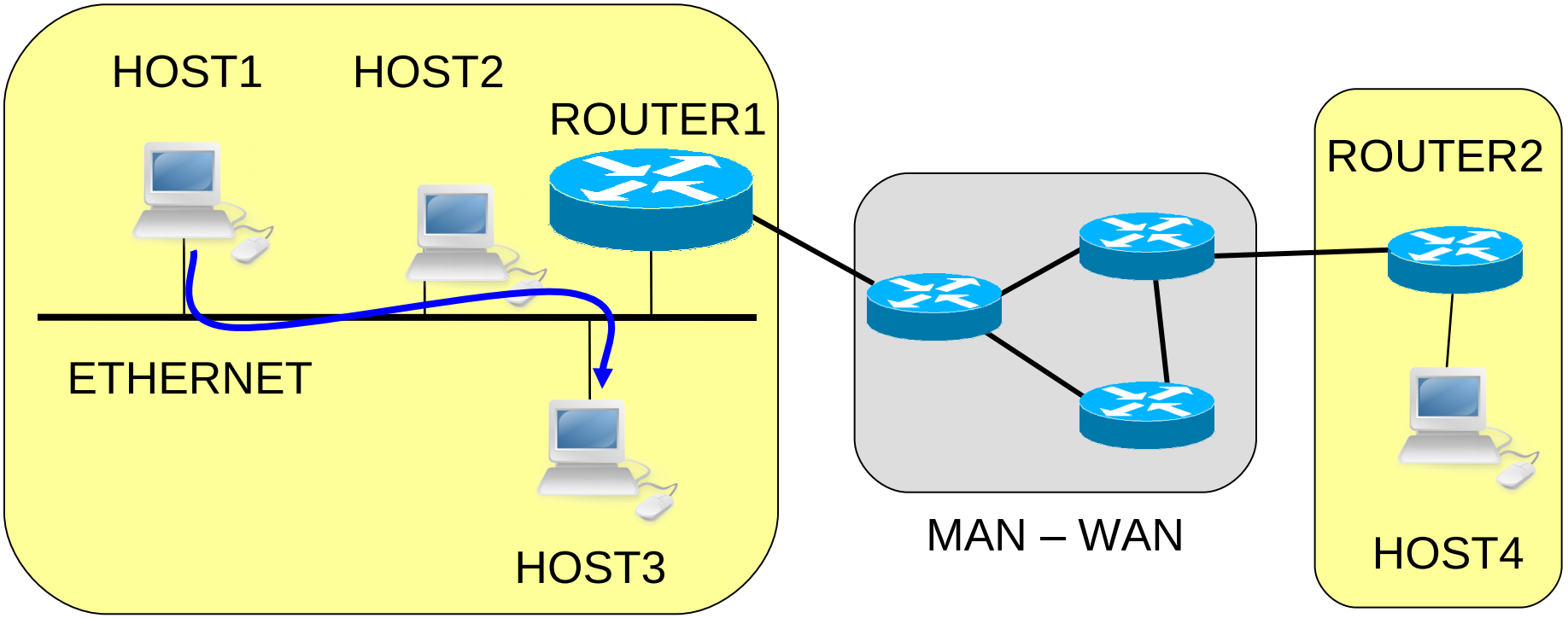


- **Routing :** scelta del percorso su cui inviare i dati
  - i **router** formano una struttura interconnessa e cooperante: i datagrammi passano dall'uno all'altro finché raggiungono quello che può consegnarli direttamente al destinatario

IP: instradamento dei datagrammi

- La consegna dei datagrammi IP all'host destinatario può avvenire in due modalità:
  - **direttamente (direct delivery) :** l'host destinatario del datagramma è sulla stessa rete di chi trasmette, quindi si spedisce il datagramma direttamente al destinatario
  - **indirettamente (indirect delivery) :** l'host destinatario del datagramma non è sulla stessa rete di chi trasmette, quindi il datagramma è inviato ad un router intermedio

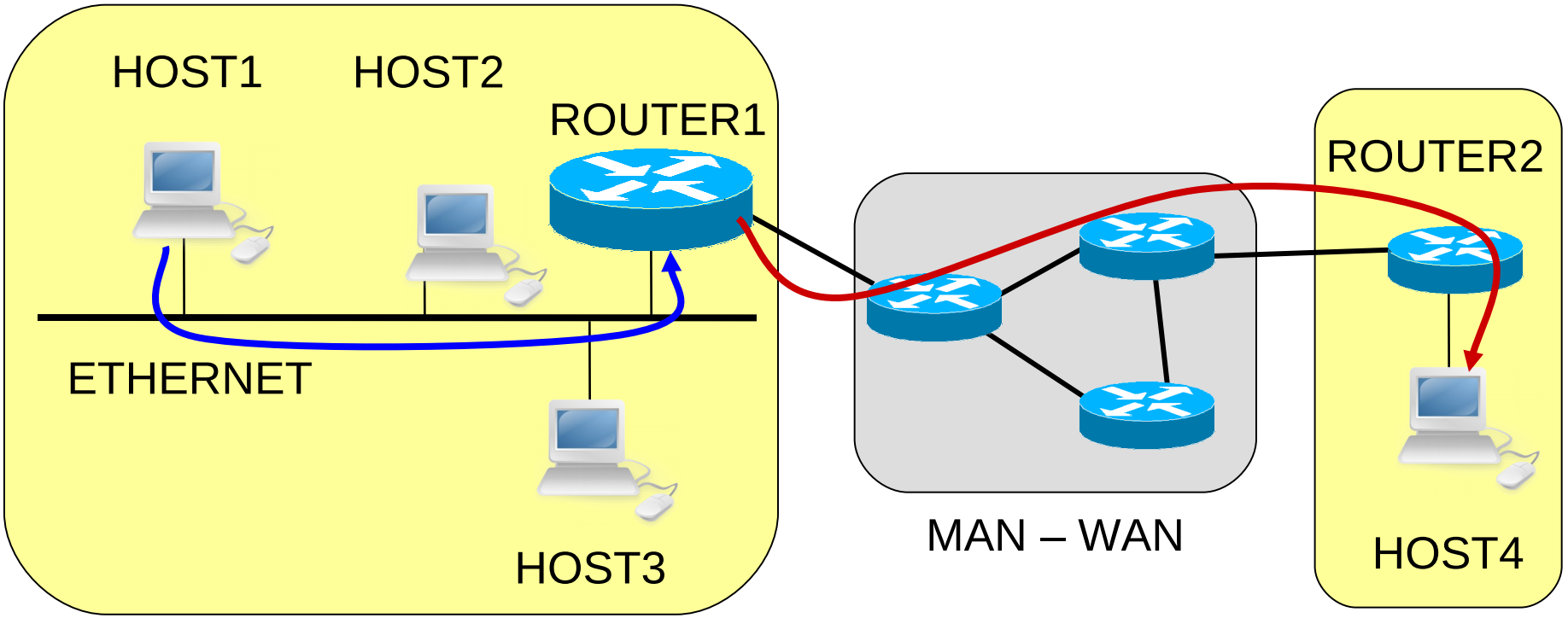
Direct delivery: da Host 1 a Host 3



ARP request HOST1 chiede l'indirizzo MAC di HOST3

ARP reply HOST3 risponde direttamente a HOST1

Indirect delivery: da Host 1 a Host 4



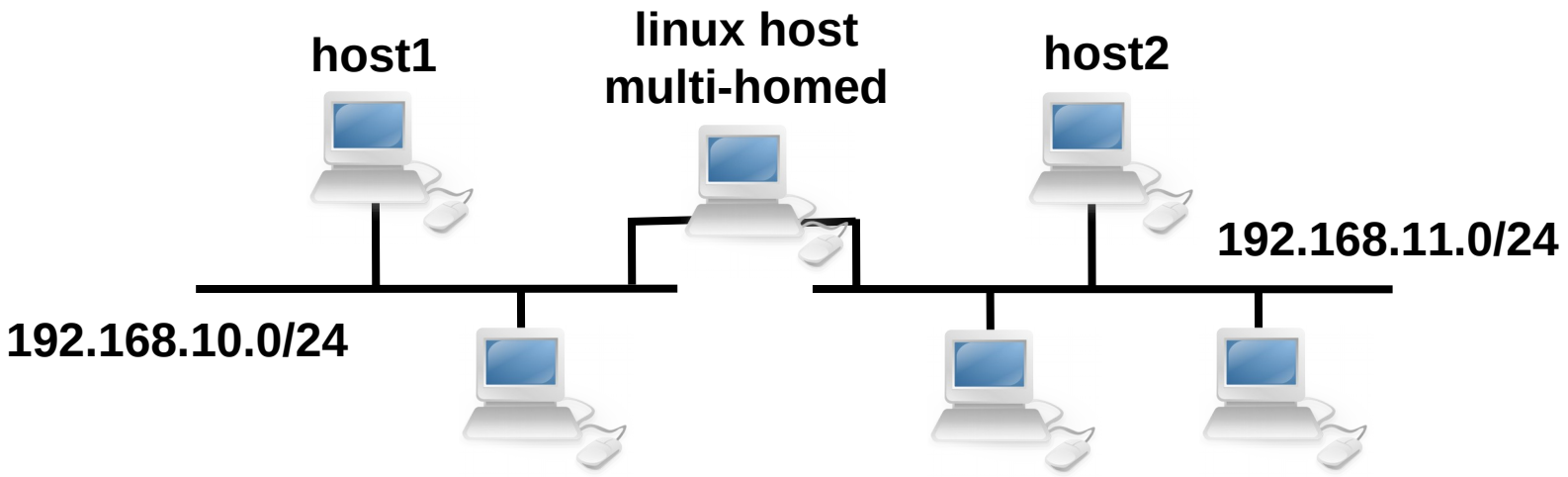
ARP request HOST1 chiede l'indirizzo MAC di ROUTER1

ARP reply ROUTER1 risponde a HOST1

MAC ADDRESS: ROUTER1	IP ADDRESS: HOST4	DATI

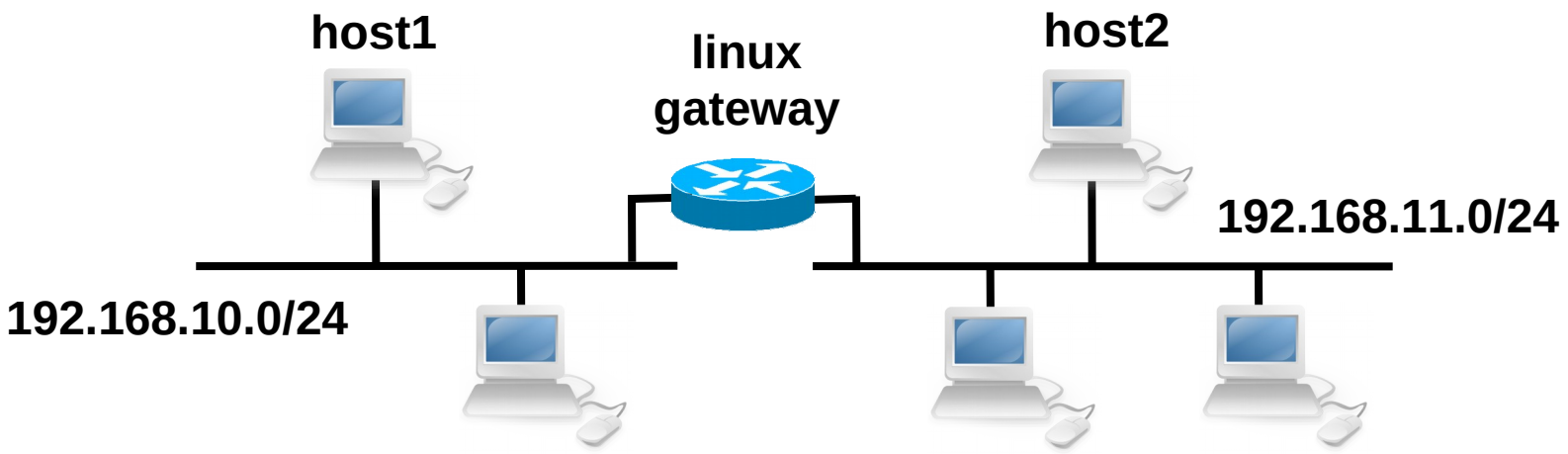
Inoltro di pacchetti IP su Linux

- Un calcolatore con s.o. Linux e interfacce di rete multiple funziona tipicamente come host multi-homed
- Esempi:
  - portatile connnesso a rete Ethernet e rete Wi-Fi
  - server con interfaccia di gestione separata dall'interfaccia usata per fornire il servizio

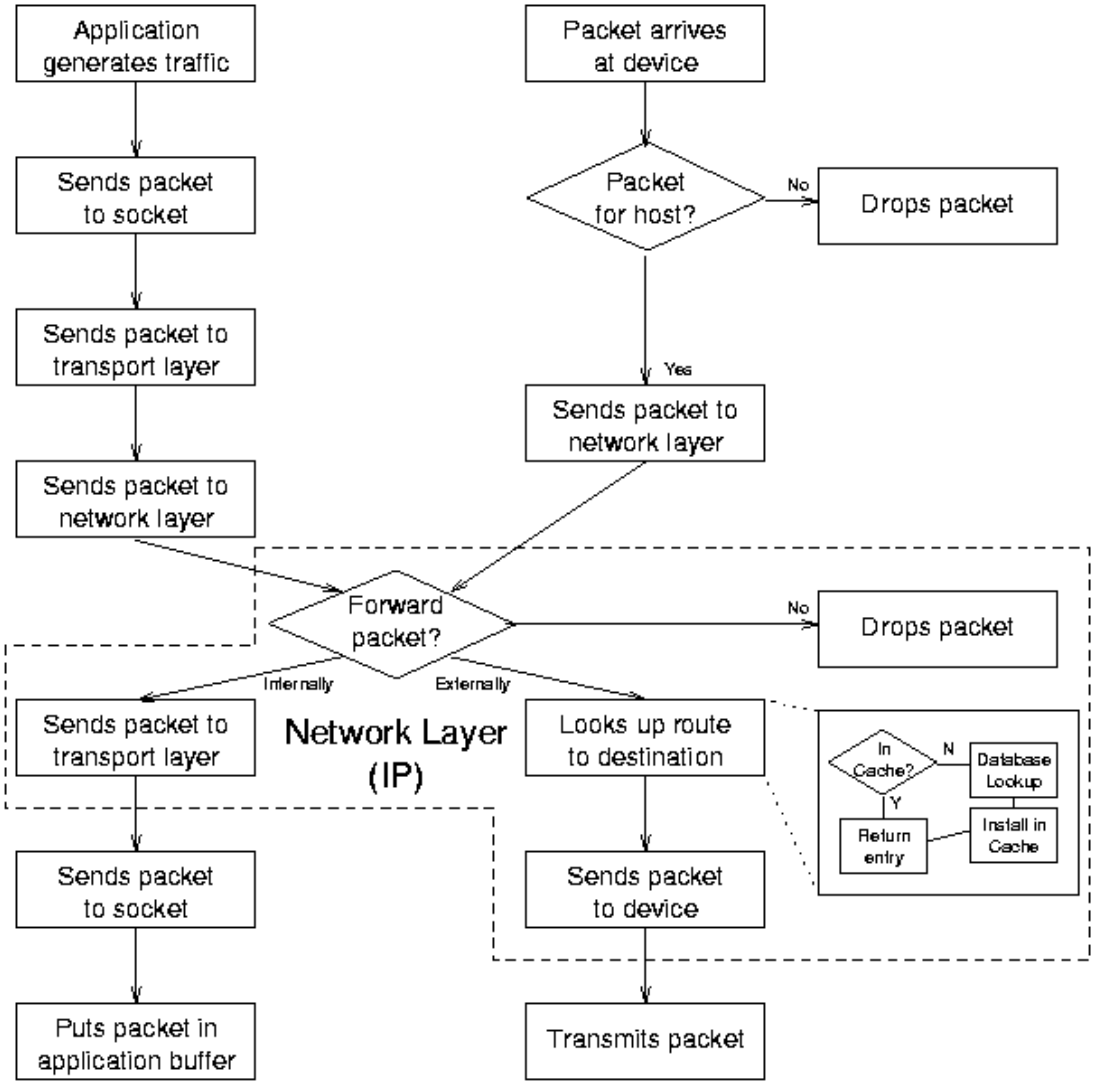


Inoltro di pacchetti IP su Linux

- Un host multi-homed Linux può essere configurato come gateway per inoltrare pacchetti tra due o più reti IP
- A questo scopo è necessario:
  - abilitare la funzione di IP forwarding
  - configurare opportunamente la tabella di instradamento per destinazioni non raggiungibili direttamente

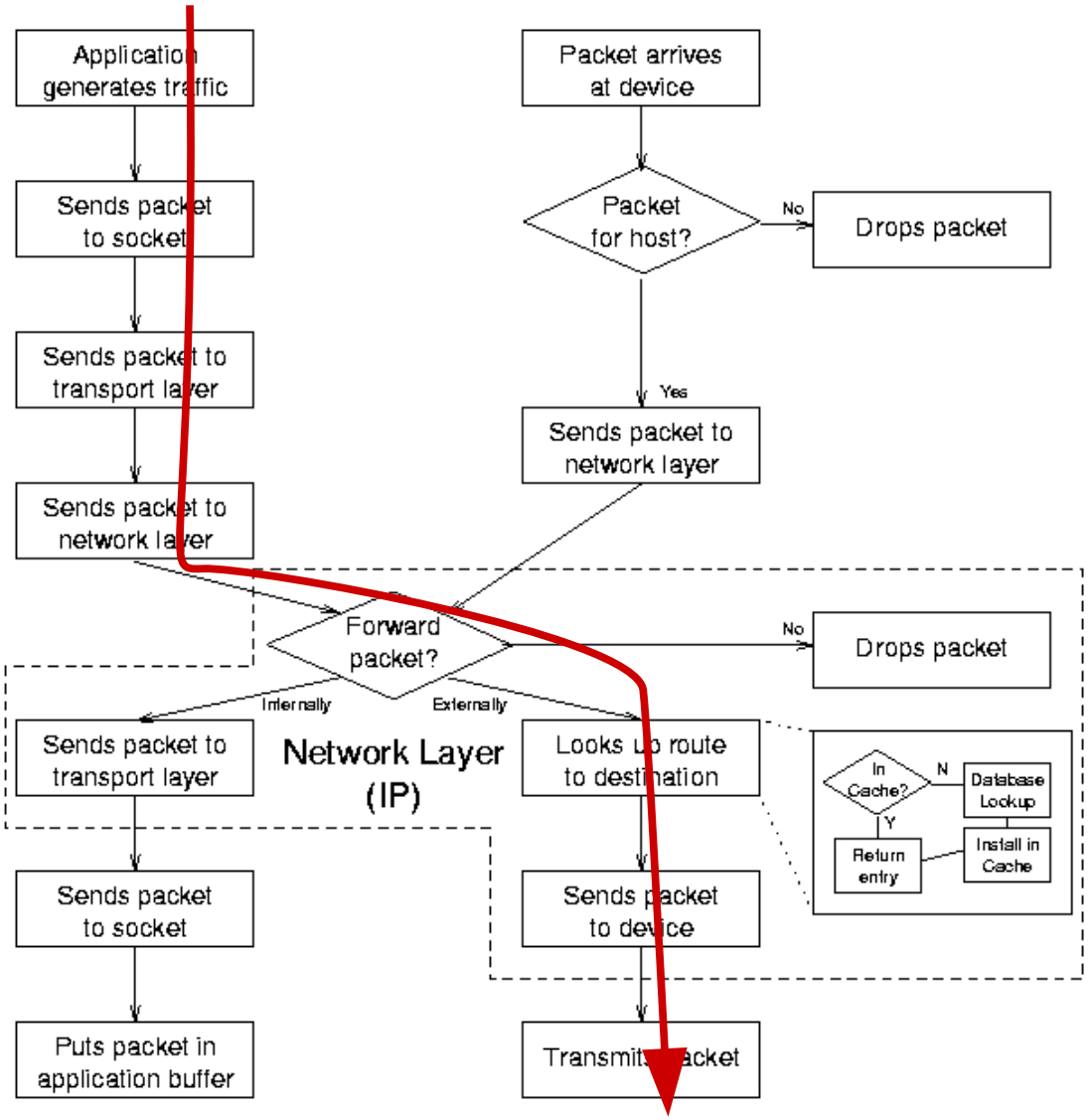


Il kernel Linux e i pacchetti IP



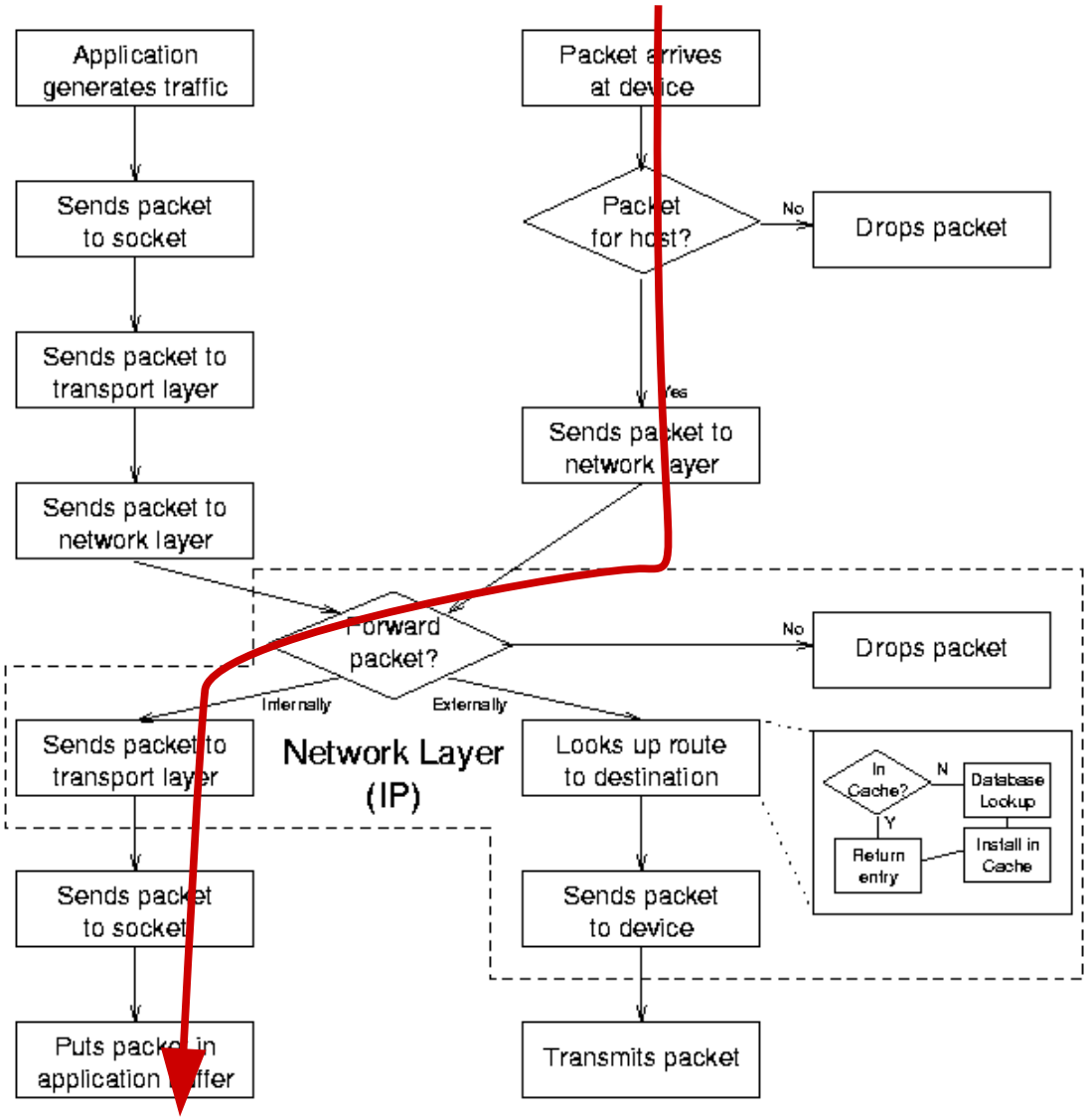
da G. Herrin, *Linux IP Networking*, http://www.cs.unh.edu/cnrg/people/gherrin/linux-net.html

Applicazione su host Linux invia pacchetti in rete



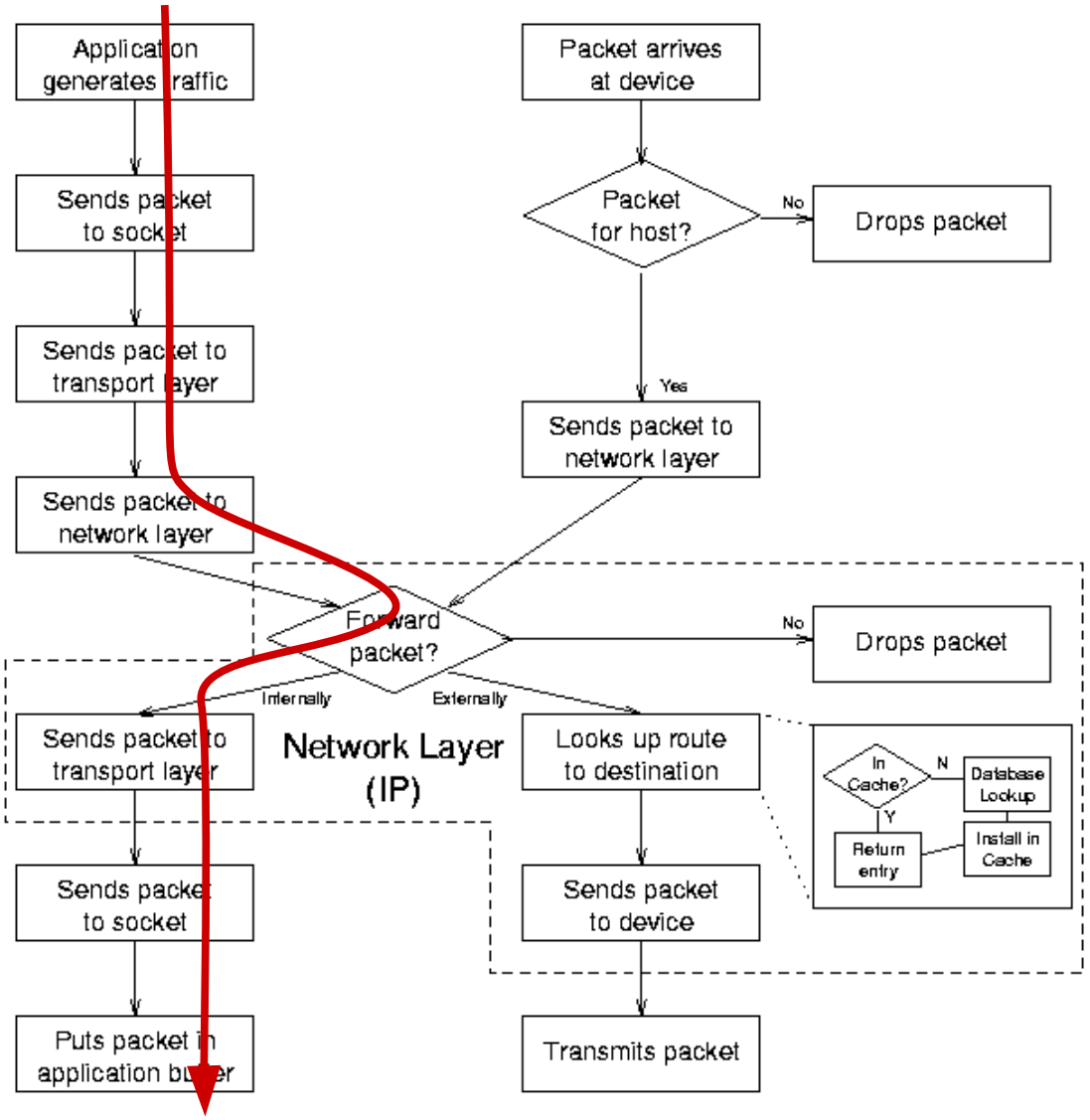
da G. Herrin, *Linux IP Networking*, http://www.cs.unh.edu/cnrg/people/gherrin/linux-net.html

Applicazione su host Linux riceve pacchetti dalla rete



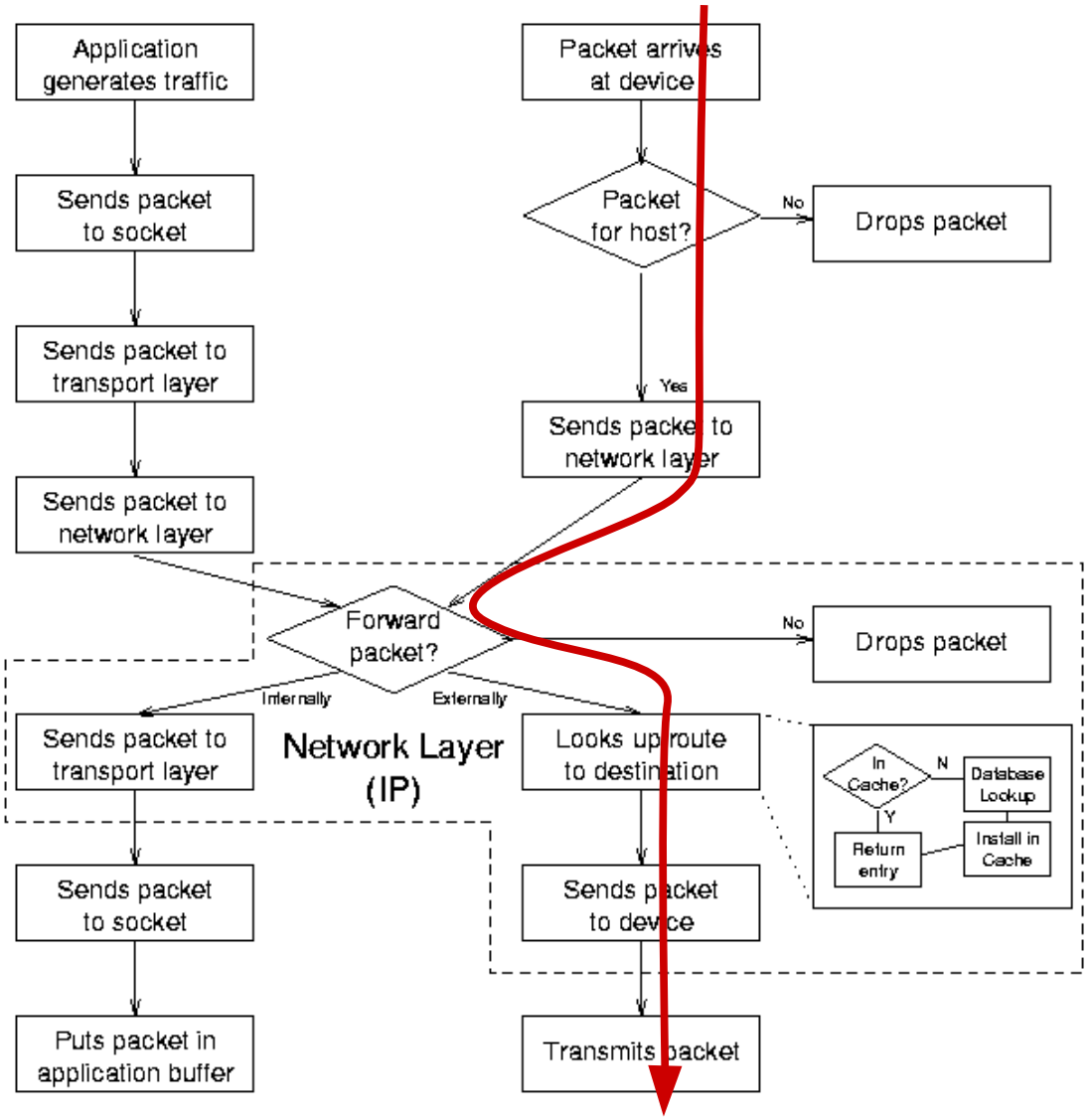
da G. Herrin, *Linux IP Networking*, http://www.cs.unh.edu/cnrg/people/gherrin/linux-net.html

Pacchetti tra due applicazioni sullo stesso host Linux



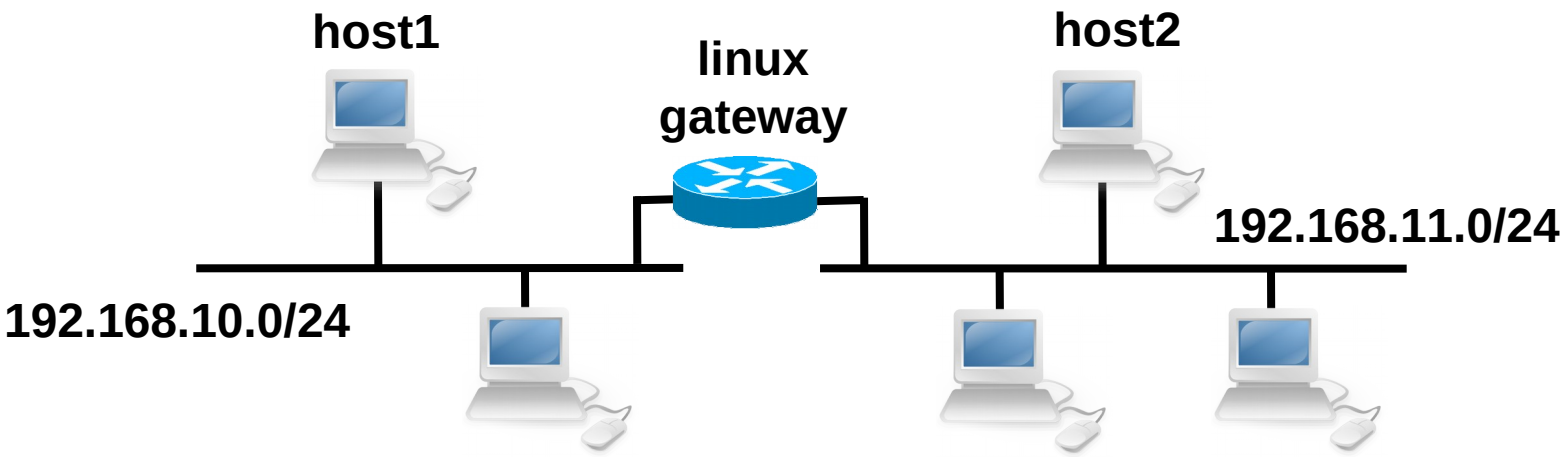
da G. Herrin, *Linux IP Networking*, http://www.cs.unh.edu/cnrg/people/gherrin/linux-net.html

Gateway Linux inoltra pacchetti tra reti



da G. Herrin, *Linux IP Networking*, http://www.cs.unh.edu/cnrg/people/gherrin/linux-net.html

IP forwarding



Il kernel deve essere abilitato all'IP forwarding:

- verifica: **sysctl net.ipv4.ip_forward**
  restituisce **1** se abilitato, **0** altrimenti
- abilitazione: **sysctl -w net.ipv4.ip_forward=1**
- disabilitazione: **sysctl -w net.ipv4.ip_forward=0**

Tabella di instradamento

- Necessaria per sapere come inoltrare i pacchetti
  - consegna **diretta** o **indiretta** tramite gateway?

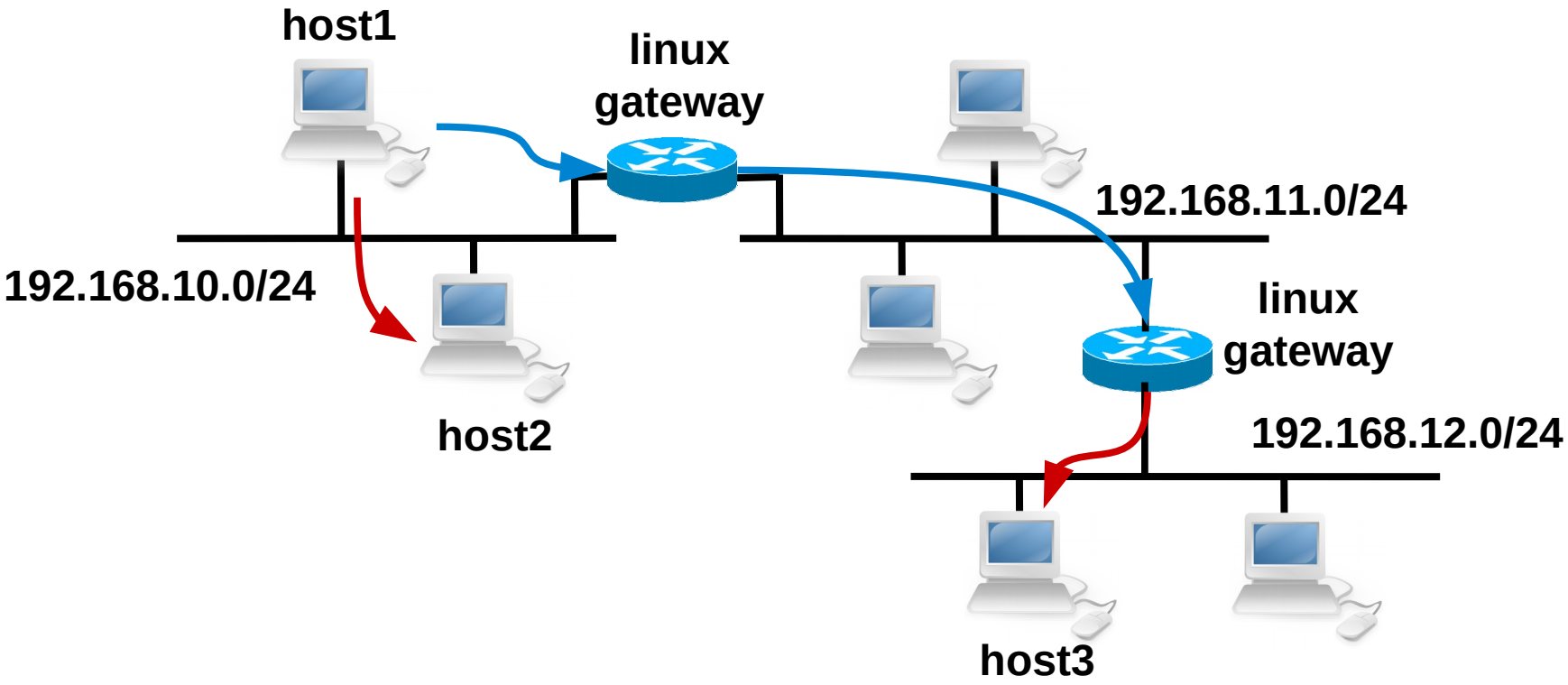


Tabella di instradamento

- Informazioni organizzate per righe
  - ogni riga rappresenta una destinazione nota
- Informazioni contenute in ciascuna riga:
  - **indirizzo di destinazione**: può essere un host o una network
  - **netmask**: utilizzata per verificare la corrispondenza dell'indirizzo di destinazione del pacchetto con la destinazione rappresentata dalla riga
  - **gateway**: indica il tipo di consegna (diretta o indiretta)
    - nel caso di consegna indiretta indica l'indirizzo del **next-hop**, cioè dell'interfaccia direttamente raggiungibile del gateway a cui inoltrare i pacchetti
  - **interfaccia di rete**: specifica quale interfaccia di rete utilizzare
  - **metrica**: specifica il "costo" di quella particolare "route"
- **Table lookup** eseguito per ogni pacchetto IP
  - host: solo quelli provenienti dagli strati superiori
  - router: anche quelli in transito

Table lookup

- La ricerca nella tabella avviene utilizzando
  - l'indirizzo IP di destinazione del pacchetto
  - l'indirizzo di destinazione e la netmask specificati in ciascuna riga della tabella
- Procedura:
  - si esegue un'operazione di **AND** bit per bit tra l'indirizzo di destinazione del pacchetto e la netmask di ciascuna riga
  - il risultato viene confrontato con la destinazione specificata nella riga stessa: se coincidono, la riga è quella giusta
  - il controllo viene effettuato a partire dalla riga che presenta una netmask con un numero maggiore di bit a uno: priorità alle route più specifiche (prima host, poi reti piccole, poi reti grandi – **longest-prefix match**)
  - una volta trovata la riga corrispondente, il lookup termina e il pacchetto viene instradato secondo la modalità specificata
  - se nessuna riga corrisponde, si usa il **default gateway**

Esempio di lookup – 1

	Destinazione	Netmask	Etc.
1	0.0.0.0	0.0.0.0	…
2	192.168.2.0	255.255.255.0	…
3	192.168.2.18	255.255.255.255	…

- Pacchetto con IP dest. = 192.168.2.18
- Confronto prima con riga 3, poi con riga 2 e poi riga 1

192.168.002.018
255.255.255.255
192.168.002.018 == 192.168.002.018

- La riga 3 è quella giusta (host specific)

Esempio di lookup – 2

	Destinazione	Netmask	Etc.
1	0.0.0.0	0.0.0.0	…
2	192.168.2.0	255.255.255.0	…
3	192.168.2.18	255.255.255.255	…

- Pacchetto con IP dest. = 192.168.2.22

```
192.168.002.022
255.255.255.255
192.168.002.022 != 192.168.002.018

192.168.002.022
255.255.255.000
192.168.002.000 == 192.168.002.000
```

- La riga 2 è quella giusta (network specific)

Esempio di lookup – 3

	Destinazione	Netmask	Etc.
1	0.0.0.0	0.0.0.0	…
2	192.168.2.0	255.255.255.0	…
3	192.168.2.18	255.255.255.255	…

- Pacchetto con IP dest. = 80.48.15.170

```
080.048.015.170
255.255.255.255
080.048.015.170 != 192.168.002.018

080.048.015.170
255.255.255.000
080.048.015.000 != 192.168.002.000

080.048.015.170
000.000.000.000
000.000.000.000 == 000.000.000.000
```

- La riga 1 è quella giusta (default gateway)

Visualizzazione tabella di routing - Windows

route print

```
C:\Users\walter>route print
===========================================================================
Interface List
 14...52 54 00 49 2e 0d ......Red Hat VirtIO Ethernet Adapter
  1...........................Software Loopback Interface 1
 11...00 00 00 00 00 00 00 e0 Microsoft ISATAP Adapter
 12...00 00 00 00 00 00 00 e0 Teredo Tunneling Pseudo-Interface
===========================================================================

IPv4 Route Table
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0    192.168.122.1  192.168.122.128     261
        127.0.0.0        255.0.0.0         On-link         127.0.0.1     306
        127.0.0.1  255.255.255.255         On-link         127.0.0.1     306
  127.255.255.255  255.255.255.255         On-link         127.0.0.1     306
    192.168.122.0    255.255.255.0         On-link  192.168.122.128     261
  192.168.122.128  255.255.255.255         On-link  192.168.122.128     261
  192.168.122.255  255.255.255.255         On-link  192.168.122.128     261
        224.0.0.0        240.0.0.0         On-link         127.0.0.1     306
        224.0.0.0        240.0.0.0         On-link  192.168.122.128     261
  255.255.255.255  255.255.255.255         On-link         127.0.0.1     306
  255.255.255.255  255.255.255.255         On-link  192.168.122.128     261
===========================================================================
Persistent Routes:
  Network Address          Netmask  Gateway Address  Metric
          0.0.0.0          0.0.0.0    192.168.122.1  Default
===========================================================================
```

Gateway = On-link → consegna diretta

Altrimenti → consegna indiretta tramite il gateway indicato

Visualizzazione tabella di routing - Mac

netstat -nr -f inet

```
iMac:~ walter$ netstat -nr -f inet
Routing tables

Internet:
Destination        Gateway            Flags        Refs      Use   Netif Expire
default            192.168.20.254     UGSc           28        0     en0
127                127.0.0.1          UCS             0        0     lo0
127.0.0.1          127.0.0.1          UH              2       26     lo0
169.254            link#4             UCS             0        0     en0
192.168.20         link#4             UCS             2        0     en0
192.168.20.5       127.0.0.1          UHS             0        0     lo0
192.168.20.10      6c:88:14:e4:6d:f4  UHLWIi          2      715     en0   1006
192.168.20.254     0:13:64:17:14:3a   UHLWIir        29        0     en0   1161
iMac:~ walter$
```

Gateway = link*#n* → consegna diretta

Gateway = 127.0.0.1 → consegna agli strati superiori

Gateway = MAC address → consegna diretta (cache ARP)

Altrimenti → consegna indiretta tramite il gateway indicato

Visualizzazione tabella di routing - Linux

route -n

```
[walter@deisnet99 ~]$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.10.254  0.0.0.0         UG    0      0        0 em1
192.168.10.0    0.0.0.0         255.255.255.0   U     1      0        0 em1
[walter@deisnet99 ~]$
```

Gateway = 0.0.0.0 → consegna diretta

Altrimenti → consegna indiretta tramite il gateway indicato

Multi-homed host

- Se sono presenti più interfacce, c'è una entry per ogni rete IP a cui si è connessi
  - necessaria per eseguire correttamente la consegna diretta

```
[root@netlab05 ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.16       0.0.0.0         255.255.255.252 U     0      0        0 virbr1
192.168.11.0    0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.122.0   0.0.0.0         255.255.255.0   U     0      0        0 virbr0
0.0.0.0         192.168.11.254  0.0.0.0         UG    0      0        0 eth0
[root@netlab05 ~]#
```

Esempio di tabella di instradamento

Router con 3 interfacce:

eth0 = 137.204.57.253/**??**
ppp0 = 10.0.0.9/**??**
ppp1 = 10.0.0.13/**??**

Completare la notazione CIDR

Riga	Destinazione	Netmask	Gateway	Metrica	Interfaccia
1	0.0.0.0	0.0.0.0	137.204.57.254	1	eth0
2	10.0.0.8	255.255.255.252	0.0.0.0	1	ppp0
3	10.0.0.12	255.255.255.252	0.0.0.0	1	ppp1
4	137.204.56.0	255.255.255.0	10.0.0.14	1	ppp1
5	137.204.57.0	255.255.255.0	0.0.0.0	1	eth0
6	137.204.58.0	255.255.254.0	10.0.0.10	1	ppp0
7	137.204.59.18	255.255.255.255	0.0.0.0	1	eth0
8	137.204.121.128	255.255.255.128	10.0.0.14	1	ppp1
9	137.204.121.128	255.255.255.128	10.0.0.10	2	ppp0
10	192.168.8.0	255.255.252.0	10.0.0.14	1	ppp1

Individuare la riga da utilizzare per ciascuno dei seguenti indirizzi di destinazione:
137.204.57.31, 137.204.56.28, 137.204.59.18, 137.204.59.131, 137.204.121.200, 10.0.0.17, 192.168.11.254

Modifica della tabella di routing su Linux

route add default gw <gateway>

aggiunge il default gateway alla tabella di routing

route add -net <dest> netmask <mask> gw <gateway>

route add -net <dest>/<prefix length> gw <gateway>

route add -host <dest> gw <gateway>

aggiunge una entry di consegna **indiretta** alla tabella di routing

route add -net <dest> netmask <mask> dev <interface>

route add -net <dest>/<prefix length> dev <interface>

route add -host <dest> dev <interface>

aggiunge una entry di consegna **diretta** alla tabella di routing

Modifica della tabella di routing su Linux

route del default

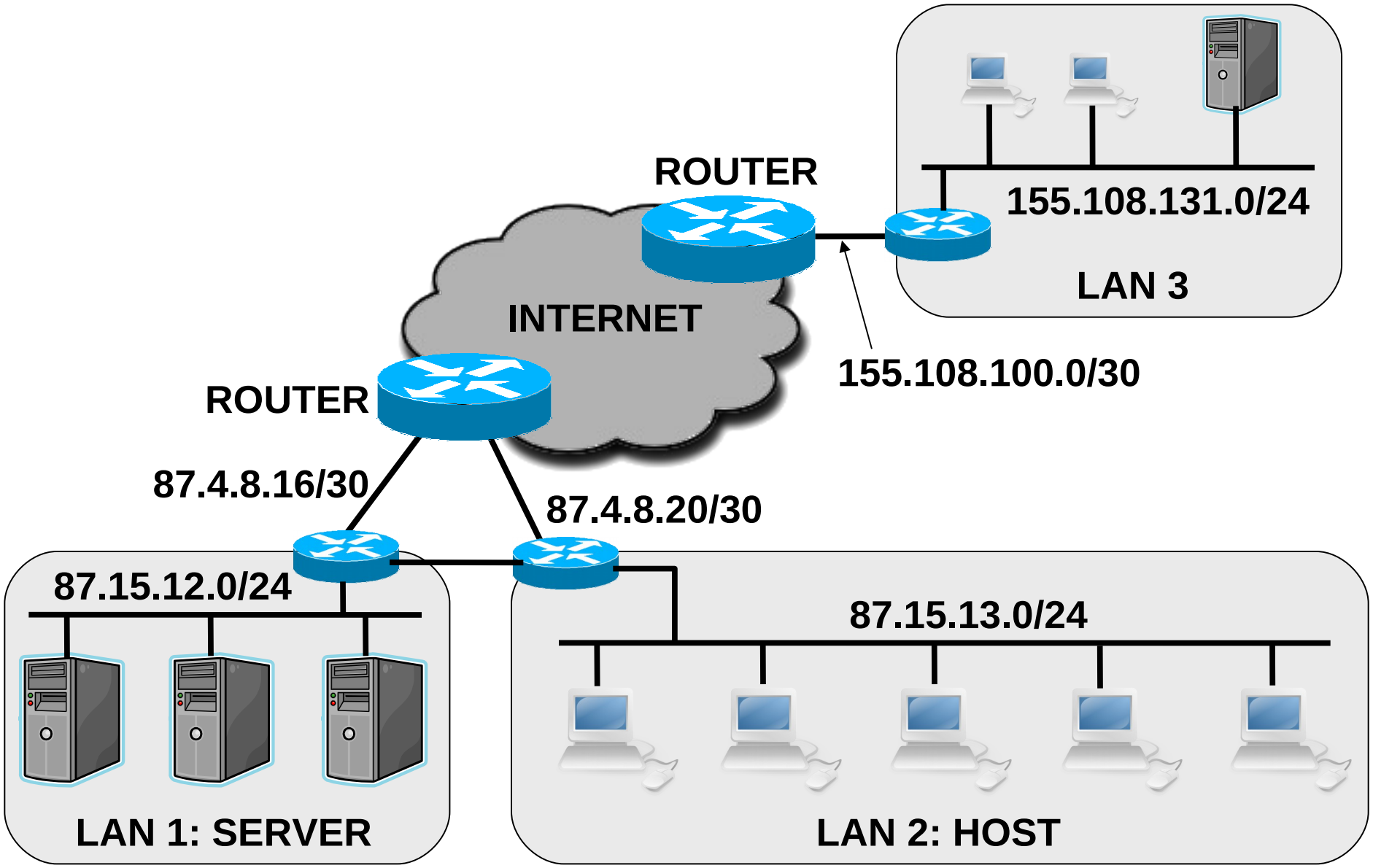
route del -net <dest> netmask <mask>

route del -net <dest>/<prefix length>

route del -host <dest>

elimina le corrispondenti entry dalla tabella

Case study: configurazione host e router



Case study: configurazione host e router

- Generico server LAN 1:
  - ifconfig eth0 87.15.12.3 netmask 255.255.255.0 broadcast 87.15.12.255
  - route add default gw 87.15.12.254

- Router LAN 1:
  - ifconfig eth0 87.15.12.254/24
  - ifconfig ppp0 87.4.8.17/30
  - ifconfig ppp1 10.0.0.1/30 ← **(da rete IP privata scelta arbitrariamente)**
  - route add default gw 87.4.8.18
  - route add -net 87.15.13.0 netmask 255.255.255.0 gw 10.0.0.2
  - sysctl -w net.ipv4.ip_forward=1

Case study: configurazione host e router

- Generico host LAN 2:
  - ifconfig eth0 87.15.13.189 netmask 255.255.255.0 broadcast 87.15.13.255
  - route add default gw 87.15.13.254

- Router LAN 2:
  - ifconfig eth0 87.15.13.254/24
  - ifconfig ppp0 87.4.8.21/30
  - ifconfig ppp1 10.0.0.2/30
  - route add default gw 87.4.8.22
  - route add -net 87.15.12.0/24 gw 10.0.0.1
  - sysctl -w net.ipv4.ip_forward=1

Case study: configurazione host e router

- Generico host/server LAN 3:
  - ifconfig eth0 155.108.131.1 netmask 255.255.255.0 broadcast 155.108.131.255
  - route add default gw 155.108.131.254

- Router LAN 3:
  - ifconfig eth0 155.108.131.254/24
  - ifconfig ppp0 155.108.100.1/30
  - route add default gw 155.108.100.2
  - sysctl -w net.ipv4.ip_forward=1