

Università degli Studi di Roma "Tor Vergata"
Laurea in Informatica

Sistemi Operativi e Reti
(modulo Reti)
a.a. 2024/2025

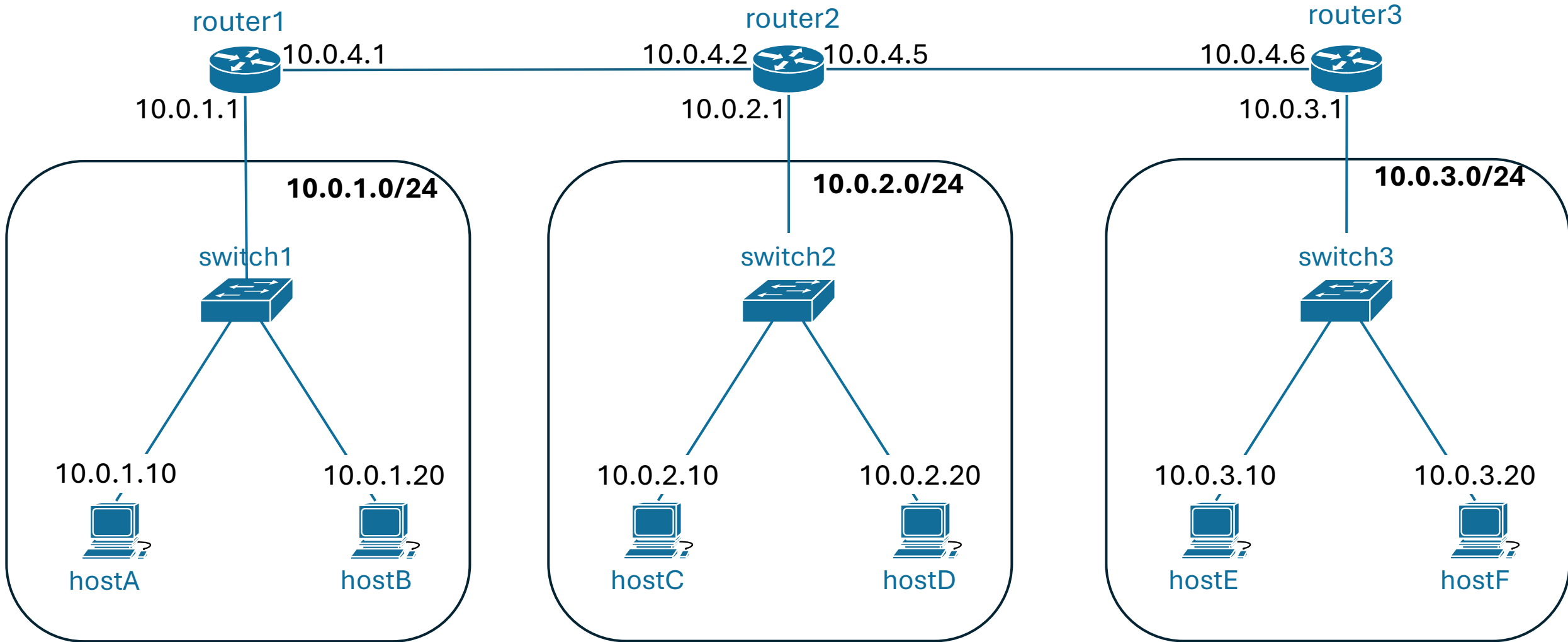
Esercitazione: virtual networking e comandi vari (parte 2)

dr. Manuel Fiorelli

manuel.fiorelli@uniroma2.it

<https://art.uniroma2.it/fiorelli>

Topologia di riferimento



Aprire una shell in un netns

Il seguente comando avvia una shell all'interno del network namespace hostA con **privilegi di root**.

```
sudo ip netns exec hostA /bin/bash
```

Per aprire una shell con i **privilegi dell'utente corrente**, occorre eseguire sudo in maniera annidata indicando l'utente originale (con l'opzione **-u**). Nell'esempio seguente trovate la dichiarazione della variabile di ambiente PS1 per personalizzare il prompt della shell in modo da far vedere che si è avviata una sessione differente.

```
PS1='\u@$(ip netns identify $$):\W] ' sudo ip netns exec hostA sudo -u $USER /bin/bash --noprofile --norc
```

In entrambi i casi si esce col comando `exit`.

Tabella di instradamento dell'host A

```
terra@hostA:~] ip route
```

```
[default via 10.0.1.1 dev eth0
```

```
[10.0.1.0/24 dev eth0 proto kernel scope link src 10.0.1.10
```

rotta di default (corrispondente al prefisso 0/0): instradamento **indiretto** attraverso (via) il router 10.0.1.1 sulla interfaccia (dev) eth0

rotta per il prefisso 10.0.1.0/24: instradamento **diretto** senza alcun router (no via) verso una sottorete direttamente connessa (scope link) all'interfaccia (dev) eth0

Svuotare la tabella ARP

Verifichiamo che la tabella ARP associata a `dev0` sia vuota:

```
ip neigh
```

Altrimenti, la svuotiamo (anteporre `sudo` se si è in una shell non privilegiata):

```
ip neigh flush dev eth0
```

Ping da Host A a Host B

```
ping -c 1 10.0.1.20
```

```
PING 10.0.1.20 (10.0.1.20) 56(84) bytes of data.
```

```
64 bytes from 10.0.1.20: icmp_seq=1 ttl=64
```

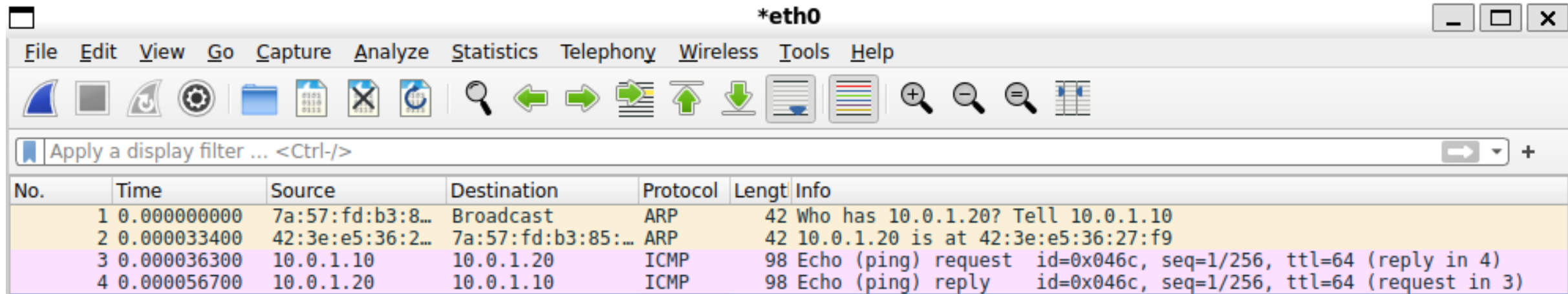
```
time=0.075 ms
```

```
--- 10.0.1.20 ping statistics ---
```

```
1 packets transmitted, 1 received, 0% packet  
loss, time 0ms
```

```
rtt min/avg/max/mdev = 0.075/0.075/0.075/0.000  
ms
```

Ping da Host A a Host B: pacchetti scambiati



The image shows a Wireshark packet capture window titled '*eth0'. The interface includes a menu bar (File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, Help) and a toolbar with various icons for packet capture and analysis. Below the toolbar is a filter bar with the text 'Apply a display filter ... <Ctrl-/>'. The main display area shows a list of four captured packets:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	7a:57:fd:b3:8...	Broadcast	ARP	42	Who has 10.0.1.20? Tell 10.0.1.10
2	0.000033400	42:3e:e5:36:2...	7a:57:fd:b3:85:...	ARP	42	10.0.1.20 is at 42:3e:e5:36:27:f9
3	0.000036300	10.0.1.10	10.0.1.20	ICMP	98	Echo (ping) request id=0x046c, seq=1/256, ttl=64 (reply in 4)
4	0.000056700	10.0.1.20	10.0.1.10	ICMP	98	Echo (ping) reply id=0x046c, seq=1/256, ttl=64 (request in 3)

1. Richiesta ARP per tradurre l'indirizzo IP da *pingare* 10.0.1.20 nell'indirizzo MAC corrispondente
2. Risposta ARP con traduzione di 10.0.1.20 in 42:3e:e5:36:27:f9
3. Messaggio ICMP Echo Request
4. Messaggio ICMP Echo Reply

Ping da Host A a Host B: Richiesta ARP

Il payload del frame Ethernet va passato all'implementazione del protocollo ARP

```
▶ Frame 1: 42 bytes on wire (336 bits) captured on eth0, id 0
  Ethernet II, Src: 7a:57:fd:b3:85:68 (7a:57:fd:b3:85:68), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
    ▶ Destination: Broadcast (ff:ff:ff:ff:ff:ff)
    ▶ Source: 7a:57:fd:b3:85:68 (7a:57:fd:b3:85:68)
    Type: ARP (0x0806)
  Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: 7a:57:fd:b3:85:68 (7a:57:fd:b3:85:68)
    Sender IP address: 10.0.1.10
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 10.0.1.20
```

Tipo e lunghezza degli indirizzi

È una richiesta

Indirizzo MAC di destinazione: indirizzo broadcast

Indirizzo MAC di sorgente: quello associato all'interfaccia eth0 dell'host A

ip link show dev eth0

4: eth0@if3:

<BROADCAST,MULTICAST,UP,LOWER_UP>

mtu 1500 qdisc noqueue state UP mode

DEFAULT group default qlen 1000

link/ether **7a:57:fd:b3:85:68** brd

ff:ff:ff:ff:ff:ff link-netnsid 0

La parte sender nella richiesta ARP si riferisce all'host A

Nella parte target, troviamo l'indirizzo IP da tradurre (e ovviamente, l'indirizzo hardware è azzerato)

Ping da Host A a Host B: Risposta ARP

Il payload del frame Ethernet va passato all'implementazione del protocollo ARP

```
▶ Frame 2: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface eth0, id 0
▼ Ethernet II, Src: 42:3e:e5:36:27:f9 (42:3e:e5:36:27:f9), Dst: 7a:57:fd:b3:85:68 (7a:57:fd:b3:85:68)
  ▶ Destination: 7a:57:fd:b3:85:68 (7a:57:fd:b3:85:68)
  ▶ Source: 42:3e:e5:36:27:f9 (42:3e:e5:36:27:f9)
  Type: ARP (0x0806)
▼ Address Resolution Protocol (reply)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: reply (2)
  Sender MAC address: 42:3e:e5:36:27:f9 (42:3e:e5:36:27:f9)
  Sender IP address: 10.0.1.20
  Target MAC address: 7a:57:fd:b3:85:68 (7a:57:fd:b3:85:68)
  Target IP address: 10.0.1.10
```

Indirizzo MAC di destinazione: quello associato all'interfaccia eth0 dell'host A

Indirizzo MAC di sorgente: quello associato all'interfaccia eth0 dell'host B

```
ip link show dev eth0
4: eth0@if3:
<BROADCAST,MULTICAST,UP,LOWER_UP>
mtu 1500 qdisc noqueue state UP mode
DEFAULT group default qlen 1000
    link/ether 7a:57:fd:b3:85:68 brd
ff:ff:ff:ff:ff:ff link-netnsid 0
```

La parte sender nella richiesta ARP si riferisce all'host B: qui si trova l'indirizzo MAC corrispondente all'indirizzo IP da tradurre

La parte target si riferisce all'host A, che aveva inviato inizialmente la richiesta

Ping hostB: ICMP Echo Request

Il payload del frame Ethernet va passato all'implementazione del protocollo IP

▶ Frame 3: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface eth0, id 0

▼ Ethernet II, Src: 7a:57:fd:b3:85:68 (7a:57:fd:b3:85:68),

▶ Destination: 42:3e:e5:36:27:f9 (42:3e:e5:36:27:f9)

▶ Source: 7a:57:fd:b3:85:68 (7a:57:fd:b3:85:68)

Type: IPv4 (0x0800)

Indirizzo MAC di destinazione: quello dell'interfaccia dell'host B

▼ Internet Protocol Version 4, Src: 10.0.1.10, Dst: 10.0.1.20

0100 = Version: 4

.... 0101 = Header Length: 20 bytes (5)

▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Total Length: 84

Identification: 0x46d1 (18129)

▶ 010. = Flags: 0x2, Don't fragment

...0 0000 0000 0000 = Fragment Offset: 0

Time to Live: 64

Protocol: ICMP (1)

Header Checksum: 0xddba [validation disabled]

[Header checksum status: Unverified]

Source Address: 10.0.1.10

Destination Address: 10.0.1.20

Indirizzo MAC sorgente: quello dell'interfaccia dell'host A

Il messaggio ICMP è incapsulato in un pacchetto IP

L'indirizzo IP sorgente è quello associato all'interfaccia dell'host A

L'indirizzo IP di destinazione è quello associato all'interfaccia dell'host B

▶ Internet Control Message Protocol

Ping da Host A a Host B: ICMP Echo Reply

Il payload del frame Ethernet va passato all'implementazione del protocollo IP

```
▶ Frame 4: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface eth0, id 0
▼ Ethernet II, Src: 42:3e:e5:36:27:f9 (42:3e:e5:36:27:f9),
  ▶ Destination: 7a:57:fd:b3:85:68 (7a:57:fd:b3:85:68)
  ▶ Source: 42:3e:e5:36:27:f9 (42:3e:e5:36:27:f9)
  Type: IPv4 (0x0800)
▼ Internet Protocol Version 4, Src: 10.0.1.20, Dst: 10.0.1.10
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 84
  Identification: 0x17a6 (6054)
  ▶ 000. .... = Flags: 0x0
  ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 64
  Protocol: ICMP (1)
  Header Checksum: 0x4ce6 [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 10.0.1.20
  Destination Address: 10.0.1.10
▶ Internet Control Message Protocol
```

Indirizzo MAC di destinazione: quello dell'interfaccia dell'host A

Indirizzo MAC sorgente: quello dell'interfaccia dell'host B

Il messaggio ICMP è incapsulato in un pacchetto IP

L'indirizzo IP sorgente è quello associato all'interfaccia dell'host B

L'indirizzo IP di destinazione è quello associato all'interfaccia dell'host A

Ping da Host A a Host B: sintesi

Il livello di rete ha trasmesso i datagrammi direttamente al destinatario utilizzando il livello di collegamento (instradamento diretto): l'indirizzo MAC di destinazione è risolvendo l'indirizzo IP del destinatario attraverso il protocollo ARP (qualora non già presente nella tabella ARP associata all'interfaccia di uscita).

Ping da Host A a Host F

```
ping -c 10.0.3.20
```

```
ping: invalid argument: '10.0.3.20'
```

```
terra@hostA:~] ping -c 1 10.0.3.20
```

```
PING 10.0.3.20 (10.0.3.20) 56(84) bytes of data.
```

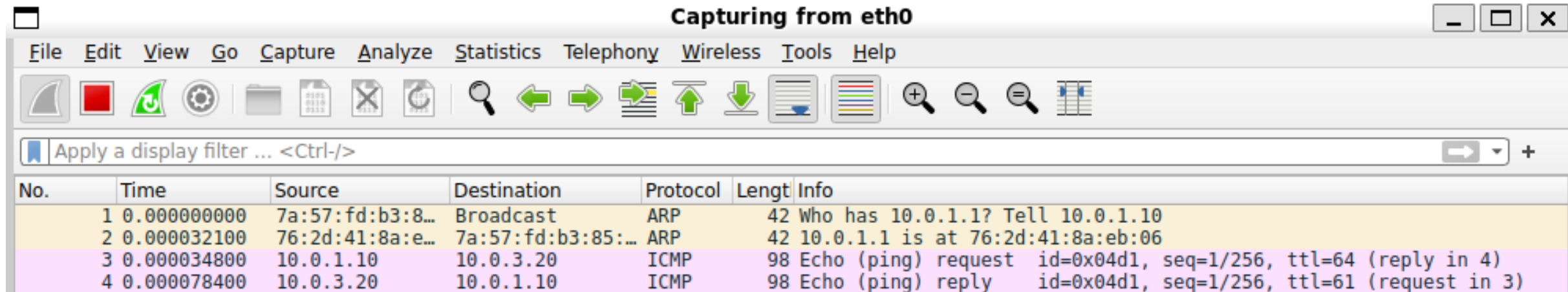
```
64 bytes from 10.0.3.20: icmp_seq=1 ttl=61  
time=0.195 ms
```

```
--- 10.0.3.20 ping statistics ---
```

```
1 packets transmitted, 1 received, 0% packet loss,  
time 0ms
```

```
rtt min/avg/max/mdev = 0.195/0.195/0.195/0.000 ms
```

Ping da Host A a Host F : pacchetti scambiati



The image shows a Wireshark packet capture window titled "Capturing from eth0". The interface includes a menu bar (File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, Help) and a toolbar with various icons for packet capture and analysis. Below the toolbar is a display filter field containing "Apply a display filter ... <Ctrl-/>". The main packet list table shows four captured packets:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	7a:57:fd:b3:8...	Broadcast	ARP	42	Who has 10.0.1.1? Tell 10.0.1.10
2	0.000032100	76:2d:41:8a:e...	7a:57:fd:b3:85:...	ARP	42	10.0.1.1 is at 76:2d:41:8a:eb:06
3	0.000034800	10.0.1.10	10.0.3.20	ICMP	98	Echo (ping) request id=0x04d1, seq=1/256, ttl=64 (reply in 4)
4	0.000078400	10.0.3.20	10.0.1.10	ICMP	98	Echo (ping) reply id=0x04d1, seq=1/256, ttl=61 (request in 3)

1. Richiesta ARP per tradurre l'indirizzo IP del router gateway 10.0.1.1 nell'indirizzo MAC corrispondente
2. Risposta ARP con traduzione di 10.0.1.1 in 76:2d:41:8a:eb:06
3. Messaggio ICMP Echo Request
4. Messaggio ICMP Echo Reply

Analizzeremo nel dettaglio solo i passi 3 e 4 perché i primi due sono analoghi a quelli visti prima.

Ping da Host A a Host F : ICMP Echo Request

Il payload del frame Ethernet va passato all'implementazione del protocollo IP

▶ Frame 3: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface eth0, id 0

▼ Ethernet II, Src: 7a:57:fd:b3:85:68 (7a:57:fd:b3:85:68),

▶ Destination: 76:2d:41:8a:eb:06 (76:2d:41:8a:eb:06)

▶ Source: 7a:57:fd:b3:85:68 (7a:57:fd:b3:85:68)

Type: IPv4 (0x0800)

Indirizzo MAC di destinazione: quello dell'interfaccia del router lato rete 10.0.1.0/24

▼ Internet Protocol Version 4, Src: 10.0.1.10, Dst: 10.0.3.20

0100 = Version: 4

.... 0101 = Header Length: 20 bytes (5)

▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Total Length: 84

Identification: 0xc684 (50820)

▶ 010. = Flags: 0x2, Don't fragment

...0 0000 0000 0000 = Fragment Offset: 0

Time to Live: 64

Protocol: ICMP (1)

Header Checksum: 0x5c07 [validation disabled]

[Header checksum status: Unverified]

Source Address: 10.0.1.10

Destination Address: 10.0.3.20

Indirizzo MAC sorgente: quello dell'interfaccia dell'host A

Il messaggio ICMP è incapsulato in un pacchetto IP

L'indirizzo IP sorgente è quello associato all'interfaccia dell'host A

L'indirizzo IP di destinazione è quello associato all'interfaccia dell'host B (NON quella del router!!!!)

▶ Internet Control Message Protocol

Ping da Host A a Host F : ICMP Echo Reply

Il payload del frame Ethernet va passato all'implementazione del protocollo IP

▶ Frame 4: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface eth0, id 0

▼ Ethernet II, Src: 76:2d:41:8a:eb:06 (76:2d:41:8a:eb:06),

▶ Destination: 7a:57:fd:b3:85:68 (7a:57:fd:b3:85:68)

▶ Source: 76:2d:41:8a:eb:06 (76:2d:41:8a:eb:06)

Type: IPv4 (0x0800)

Indirizzo MAC di destinazione: quello dell'interfaccia del router lato rete 10.0.1.0/24

▼ Internet Protocol Version 4, Src: 10.0.3.20, Dst: 10.0.1.10

0100 = Version: 4

.... 0101 = Header Length: 20 bytes (5)

▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECN)

Total Length: 84

Identification: 0x8a53 (35411)

▶ 000. = Flags: 0x0

...0 0000 0000 0000 = Fragment Offset: 0

Time to Live: 61

Protocol: ICMP (1)

Header Checksum: 0xdb38 [validation disabled]

[Header checksum status: Unverified]

Source Address: 10.0.3.20

Destination Address: 10.0.1.10

Indirizzo MAC di destinazione: quello dell'interfaccia del router lato rete 10.0.1.0/24

Il messaggio ICMP è incapsulato in un pacchetto IP

L'indirizzo IP sorgente è quello associato all'interfaccia dell'host B (NON quella del router!!!)

L'indirizzo IP di destinazione è quello associato all'interfaccia dell'host A

▶ Internet Control Message Protocol

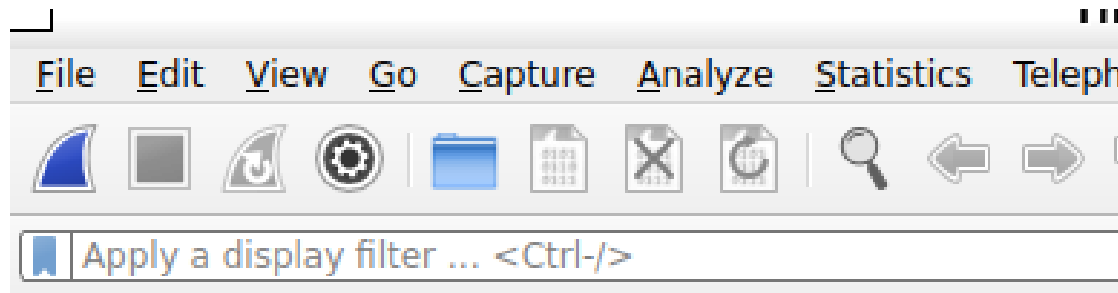
Ping da Host A a Host F : sintesi

Il livello di rete ha instradato i datagrammi in maniera indiretta attraverso il gateway:

- Livello di rete: gli indirizzi IP sorgente e destinazione sono quelli dell'host A e dell'host F
- Livello di collegamento: relativamente al primo collegamento tra host A e router 1, gli indirizzi MAC sorgente e destinazione sono quelli dell'host A e del router 1

Ping da Host A a Host F : router 1

Occorre aprire una shell nel **network namespace router1**



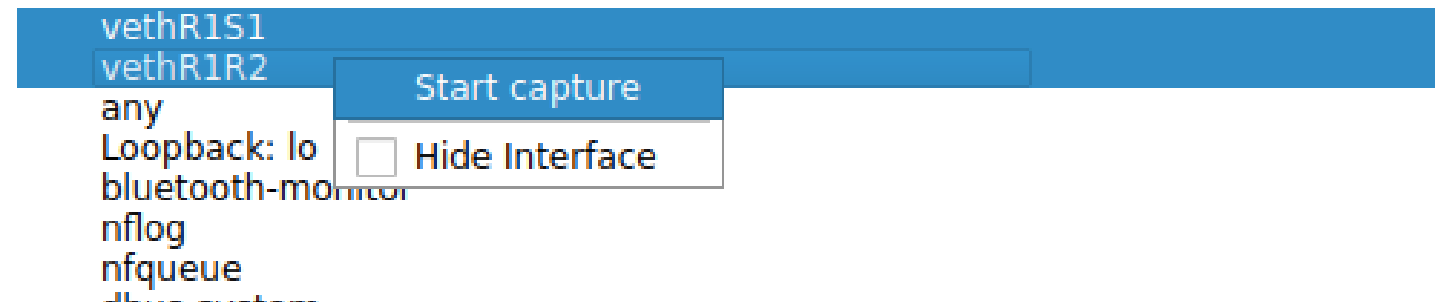
Selezionare più interfacce: basta clicca tenendo premuto CTRL.

Successivamente, avviare la cattura selezionando la voce "Start capture" nel menu contestuale aperto col tasto destro

Welcome to Wireshark

Capture

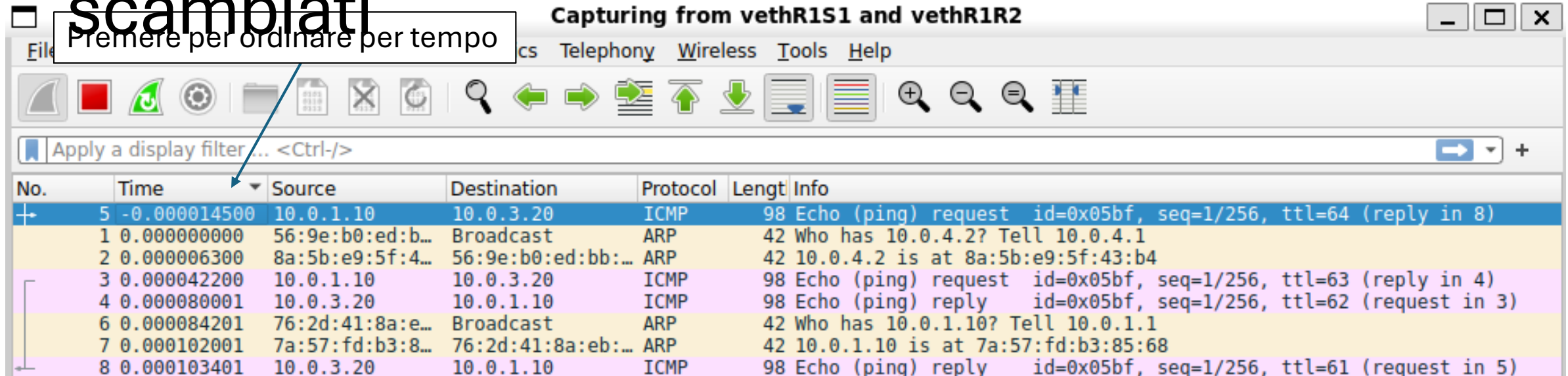
...using this filter:



Ping da Host A a Host F: router 1: pacchetti

scambiati

Premere per ordinare per tempo



The image shows a Wireshark packet capture window titled "Capturing from vethR1S1 and vethR1R2". The packet list shows 8 packets. A blue arrow points from the text "Premere per ordinare per tempo" to the "Time" column header. The packets are as follows:

No.	Time	Source	Destination	Protocol	Length	Info
5	-0.000014500	10.0.1.10	10.0.3.20	ICMP	98	Echo (ping) request id=0x05bf, seq=1/256, ttl=64 (reply in 8)
1	0.000000000	56:9e:b0:ed:b...	Broadcast	ARP	42	Who has 10.0.4.2? Tell 10.0.4.1
2	0.000006300	8a:5b:e9:5f:4...	56:9e:b0:ed:bb:...	ARP	42	10.0.4.2 is at 8a:5b:e9:5f:43:b4
3	0.000042200	10.0.1.10	10.0.3.20	ICMP	98	Echo (ping) request id=0x05bf, seq=1/256, ttl=63 (reply in 4)
4	0.000080001	10.0.3.20	10.0.1.10	ICMP	98	Echo (ping) reply id=0x05bf, seq=1/256, ttl=62 (request in 3)
6	0.000084201	76:2d:41:8a:e...	Broadcast	ARP	42	Who has 10.0.1.10? Tell 10.0.1.1
7	0.000102001	7a:57:fd:b3:8...	76:2d:41:8a:eb:...	ARP	42	10.0.1.10 is at 7a:57:fd:b3:85:68
8	0.000103401	10.0.3.20	10.0.1.10	ICMP	98	Echo (ping) reply id=0x05bf, seq=1/256, ttl=61 (request in 5)

1. Echo request destinata all'host F e ricevuta attraverso l'interfaccia vethS1R1
2. Richiesta ARP per tradurre l'indirizzo IP del next hop 10.0.4.2
3. Risposta ARP con traduzione di 10.0.4.2 in 8a:5b:e9:5f:43:b4
4. Echo request destinata all'host F inoltrata a 10.0.4.2 attraverso l'interfaccia vethR1R2
5. Echo reply destinata all'host A e ricevuta attraverso l'interfaccia vethR1R2
6. Richiesta ARP per tradurre l'indirizzo IP dell'host A 10.0.1.10
7. Risposta ARP con traduzione di 10.0.1.1 in 7a:57:fd:b3:85:68
8. Echo request destinata all'host A trasmessa al destinatario attraverso l'interfaccia vethR1R2

Ping hostF: router 1: inoltro Echo request (1)

Capturing from vethR1S1 and vethR1R2

No.	Time	Source	Destination	Protocol	Length	Info
5	-0.000014500	10.0.1.10	10.0.3.20	ICMP	98	Echo (ping) request id=0x05bf, seq=1/256, ttl=64 (reply in 8)
1	0.000000000	56:9e:b0:ed:b...	Broadcast	ARP	42	Who has 10.0.4.2? Tell 10.0.4.1
2	0.000006300	8a:5b:e9:5f:4...	56:9e:b0:ed:bb:...	ARP	42	10.0.4.2 is at 8a:5b:e9:5f:43:b4
3	0.000042200	10.0.1.10	10.0.3.20	ICMP	98	Echo (ping) request id=0x05bf, seq=1/256, ttl=63 (reply in 4)
4	0.000080001	10.0.3.20	10.0.1.10	ICMP	98	Echo (ping) reply id=0x05bf, seq=1/256, ttl=62 (request in 3)
6	0.000084201	76:2d:41:8a:e...	Broadcast	ARP	42	Who has 10.0.1.10? Tell 10.0.1.1
7	0.000102001	7a:57:fd:b3:8...	76:2d:41:8a:eb:...	ARP	42	10.0.1.10 is at 7a:57:fd:b3:85:68
8	0.000103401	10.0.3.20	10.0.1.10	ICMP	98	Echo (ping) reply id=0x05bf, seq=1/256, ttl=61 (request in 5)

L'indirizzo MAC sorgente è quello dell'interfaccia di A

Il router ha ricevuto il messaggio Echo Request attraverso il collegamento con lo switch

- ▶ Frame 5: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface vethR1S1, id 0
- ▶ Ethernet II, Src: 7a:57:fd:b3:85:68 (7a:57:fd:b3:85:68), Dst: 76:2d:41:8a:eb:06 (76:2d:41:8a:eb:06)
- ▶ Internet Protocol Version 4, Src: 10.0.1.10, Dst: 10.0.3.20
- ▶ Internet Control Message Protocol

L'indirizzo IP sorgente è quello dell'host A

L'indirizzo IP sorgente è quello dell'host F

L'indirizzo MAC di destinazione è quello dell'interfaccia del router lato rete 10.0.1.0/24

Ping da Host A a Host F: router 1: inoltro Echo

request (2)

Capturing from vethR1S1 and vethR1R2

No.	Time	Source	Destination	Protocol	Length	Info
5	-0.000014500	10.0.1.10	10.0.3.20	ICMP	98	Echo (ping) request id=0x05bf, seq=1/256, ttl=64 (reply in 8)
1	0.000000000	56:9e:b0:ed:b...	Broadcast	ARP	42	Who has 10.0.4.2? Tell 10.0.4.1
2	0.0000006300	8a:5b:e9:5f:4...	56:9e:b0:ed:bb:...	ARP	42	10.0.4.2 is at 8a:5b:e9:5f:43:b4
3	0.000042200	10.0.1.10	10.0.3.20	ICMP	98	Echo (ping) request id=0x05bf, seq=1/256, ttl=63 (reply in 4)
4	0.000080001	10.0.3.20	10.0.1.10	ICMP	98	Echo (ping) reply id=0x05bf, seq=1/256, ttl=62 (request in 3)
6	0.000084201	76:2d:41:8a:e...	Broadcast	ARP	42	Who has 10.0.1.10? Tell 10.0.1.1
7	0.000102001	7a:57:fd:b3:8...	76:2d:41:8a:eb:...	ARP	42	10.0.1.10 is at 7a:57:fd:b3:85:68
8	0.000103401	10.0.3.20	10.0.1.10	ICMP	98	Echo (ping) reply id=0x05bf, seq=1/256, ttl=61 (request in 5)

Sul router 1:

```
$ ip route
```

```
10.0.1.0/24 dev vethR1S1 proto kernel scope link src 10.0.1.1
```

```
10.0.2.0/24 via 10.0.4.2 dev vethR1R2
```

```
10.0.3.0/24 via 10.0.4.2 dev vethR1R2
```

```
10.0.4.0/30 dev vethR1R2 proto kernel scope link src 10.0.4.1
```

```
$ ip route get 10.0.3.20
```

```
10.0.3.20 via 10.0.4.2 dev vethR1R2 src 10.0.4.1 uid 1000  
cache
```

I datagrammi destinati a 10.0.3.20 devono essere inoltrati a 10.0.4.2 attraverso l'interfaccia vethR1R2

Ping da Host A a Host F: router 1: inoltrato Echo

request (3)

Capturing from vethR1S1 and vethR1R2

No.	Time	Source	Destination	Protocol	Length	Info
5	-0.000014500	10.0.1.10	10.0.3.20	ICMP	98	Echo (ping) request id=0x05bf, seq=1/256, ttl=64 (reply in 8)
1	0.000000000	56:9e:b0:ed:b...	Broadcast	ARP	42	Who has 10.0.4.2? Tell 10.0.4.1
2	0.000006300	8a:5b:e9:5f:4...	56:9e:b0:ed:bb:...	ARP	42	10.0.4.2 is at 8a:5b:e9:5f:43:b4
3	0.000042200	10.0.1.10	10.0.3.20	ICMP	98	Echo (ping) request id=0x05bf, seq=1/256, ttl=63 (reply in 4)
4	0.000080001	10.0.3.20	10.0.1.10	ICMP	98	Echo (ping) reply id=0x05bf, seq=1/256, ttl=62 (request in 3)
6	0.000084201	76:2d:41:8a:e...	Broadcast	ARP	42	Who has 10.0.1.10? Tell 10.0.1.1
7	0.000102001	7a:57:fd:b3:8...	76:2d:41:8a:eb:...	ARP	42	10.0.1.10 is at 7a:57:fd:b3:85:68
8	0.000103401	10.0.3.20	10.0.1.10	ICMP	98	Echo (ping) reply id=0x05bf, seq=1/256, ttl=61 (request in 5)

Traduzione dell'indirizzo IP 10.0.4.2 in indirizzo MAC (se non già presente nella tabella ARP)

Ping da Host A a Host F: router 1: inoltro Echo

request (4)

Capturing from vethR1S1 and vethR1R2

No.	Time	Source	Destination	Protocol	Length	Info
5	-0.000014500	10.0.1.10	10.0.3.20	ICMP	98	Echo (ping) request id=0x05bf, seq=1/256, ttl=64 (reply in 8)
1	0.000000000	56:9e:b0:ed:b...	Broadcast	ARP	42	Who has 10.0.4.2? Tell 10.0.4.1
2	0.0000006300	8a:5b:e9:5f:4...	56:9e:b0:ed:bb:...	ARP	42	10.0.4.2 is at 8a:5b:e9:5f:43:b4
3	0.0000042200	10.0.1.10	10.0.3.20	ICMP	98	Echo (ping) request id=0x05bf, seq=1/256, ttl=63 (reply in 4)
4	0.0000080001	10.0.3.20	10.0.1.10	ICMP	98	Echo (ping) reply id=0x05bf, seq=1/256, ttl=62 (request in 3)
6	0.0000084201	76:2d:41:8a:e...	Broadcast	ARP	42	Who has 10.0.1.10? Tell 10.0.1.1
7	0.000102001	7a:57:fd:b3:8...	76:2d:41:8a:eb:...	ARP	42	10.0.1.10 is at 7a:57:fd:b3:85:68
8	0.000103401	10.0.3.20	10.0.1.10	ICMP	98	Echo (ping) reply id=0x05bf, seq=1/256, ttl=61 (request in 5)

L'indirizzo MAC sorgente è quello dell'interfaccia vethR1R2

Il router 1 inoltra il messaggio ICMP Echo request attraverso il collegamento vethR1R2

▶ Frame 3: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface vethR1R2, id 1
▶ Ethernet II, Src: 56:9e:b0:ed:bb:6f (56:9e:b0:ed:bb:6f), Dst: 8a:5b:e9:5f:43:b4 (8a:5b:e9:5f:43:b4)
▶ Internet Protocol Version 4, Src: 10.0.1.10, Dst: 10.0.3.20
▶ Internet Control Message Protocol

L'indirizzo IP sorgente è quello dell'host A

L'indirizzo IP sorgente è quello dell'host F

L'indirizzo MAC di destinazione è quello dell'interfaccia con cui il router 2 è collegato al router 1

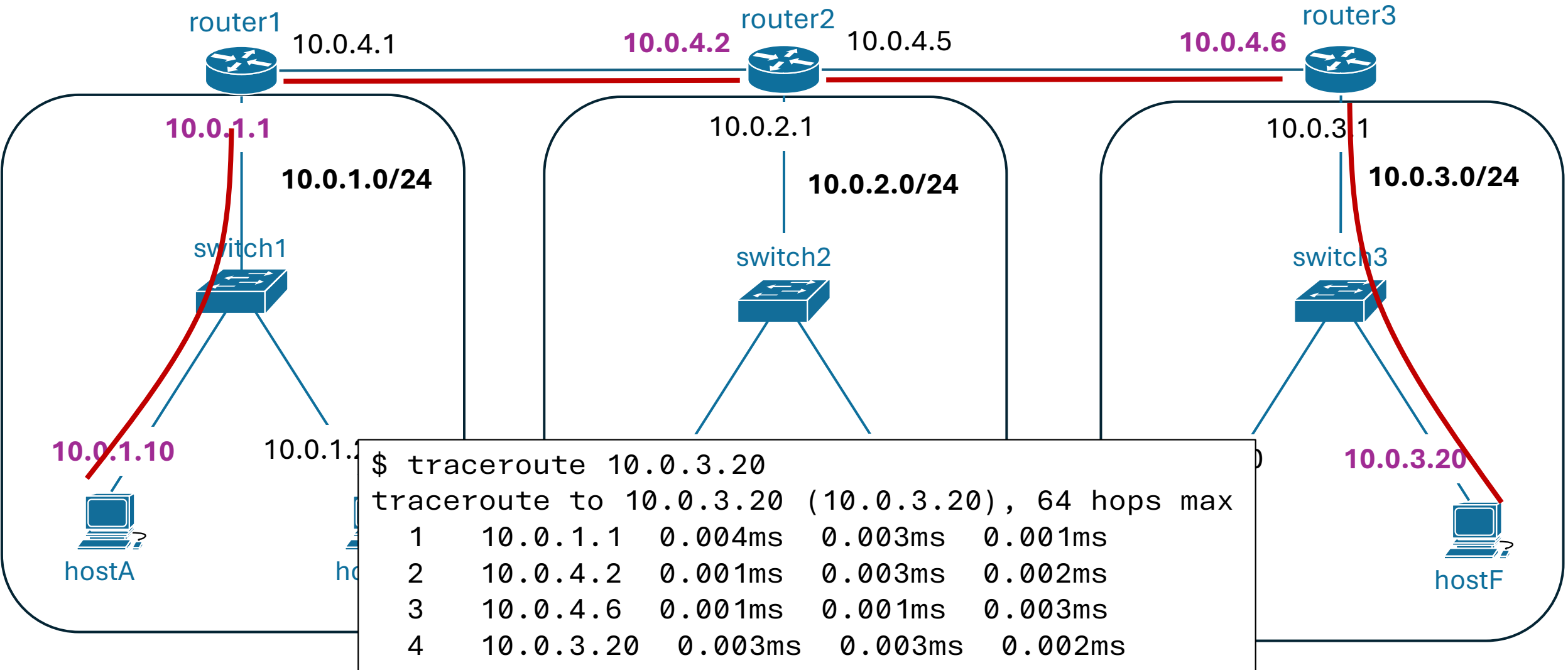
Ping da Host A a Host F: router 1: sintesi

Non descriviamo l'inoltro del messio ICMP Echo reply, perché analogo.

Sottolineiamo che l'inoltro a livello di rete:

- Non altera gli indirizzi IP sorgente e destinazione
- Agisce sugli indirizzi MAC a livello dei collegamento:
 - Sorgente: interfaccia del nodo precedente al router nel percorso dalla sorgente IP alla destinazione IP
 - Destinazione: interfaccia del nodo successivo al router nel percorso dalla sorgente IP alla destinazione IP

Traceroute da Host A a Host F



```
$ traceroute 10.0.3.20
traceroute to 10.0.3.20 (10.0.3.20), 64 hops max
 1  10.0.1.1  0.004ms  0.003ms  0.001ms
 2  10.0.4.2  0.001ms  0.003ms  0.002ms
 3  10.0.4.6  0.001ms  0.001ms  0.003ms
 4  10.0.3.20 0.003ms  0.003ms  0.002ms
```

Arping da Host A a Host B

Assicuriamoci che la tabella ARP associate a eth0 sia vuota:

```
$ sudo ip neigh flush dev eth0  
$ sudo ip neigh show dev eth0
```

Usiamo arping (dopo averlo installato, se necessario) per inviare una richiesta ARP

```
$ sudo arping -c 1 10.0.1.20  
ARPING 10.0.1.20  
42 bytes from 42:3e:e5:36:27:f9 (10.0.1.20): index=0  
time=15.900 usec
```

```
--- 10.0.1.20 statistics ---  
1 packets transmitted, 1 packets received,    0% unanswered  
(0 extra)  
rtt min/avg/max/std-dev = 0.016/0.016/0.016/0.000 ms
```

hostC

hostD

hostE

hostF

router1
10.0.4.1

10.0.1.1

switch1

10.0.1.10

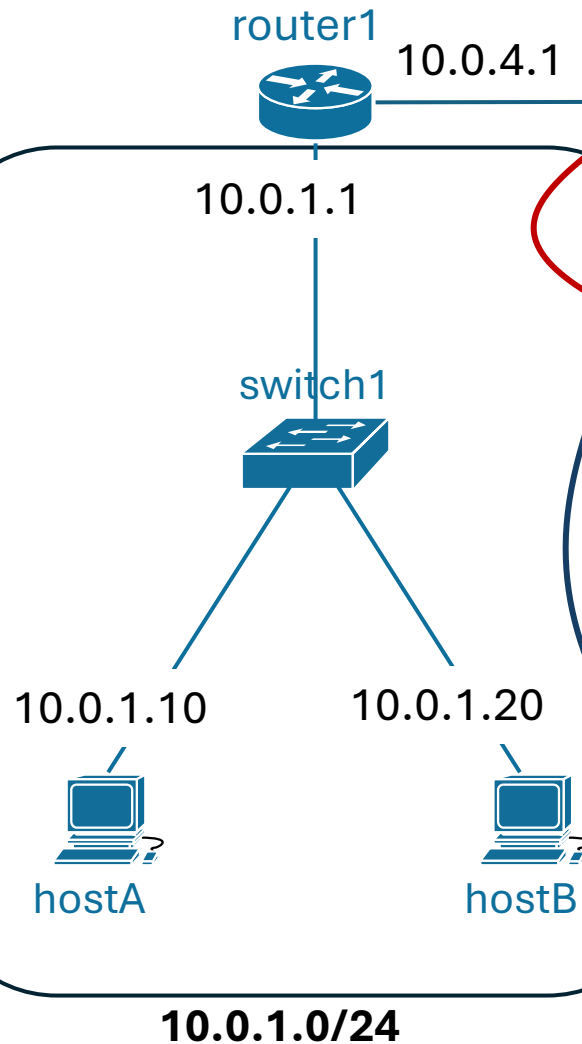
10.0.1.20

hostA

hostB

10.0.1.0/24

Argping da Host A a Host B



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	7a:57:fd:b3:8...	Broadcast	ARP	58	Who has 10.0.1.20? Tell 10.0.1.10
2	0.000106000	42:3e:e5:36:2...	7a:57:fd:b3:85:...	ARP	42	10.0.1.20 is at 42:3e:e5:36:27:f9

▶ Frame 1: 58 bytes on wire (464 bits), 58 bytes captured (464 bits) on interface eth0, id 0

▶ Ethernet II, Src: 7a:57:fd:b3:85:68 (7a:57:fd:b3:85:68), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

▶ Destination: Broadcast (ff:ff:ff:ff:ff:ff)

▶ Source: 7a:57:fd:b3:85:68 (7a:57:fd:b3:85:68)

▶ Type: ARP (0x0806)

▶ Trailer: 00000000000000000000000000000000

▼ Address Resolution Protocol (request)

Hardware type: Ethernet (1)

Protocol type: IPv4 (0x0800)

Hardware size: 6

Protocol size: 4

Opcode: request (1)

Sender MAC address: 7a:57:fd:b3:85:68 (7a:57:fd:b3:85:68)

Sender IP address: 10.0.1.10

Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)

Target IP address: 10.0.1.20

▶ Frame 2: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface eth0, id 0

▶ Ethernet II, Src: 42:3e:e5:36:27:f9 (42:3e:e5:36:27:f9), Dst: 7a:57:fd:b3:85:68 (7a:57:fd:b3:85:68)

▶ Destination: 7a:57:fd:b3:85:68 (7a:57:fd:b3:85:68)

▶ Source: 42:3e:e5:36:27:f9 (42:3e:e5:36:27:f9)

▶ Type: ARP (0x0806)

▼ Address Resolution Protocol (reply)

Hardware type: Ethernet (1)

Protocol type: IPv4 (0x0800)

Hardware size: 6

Protocol size: 4

Opcode: reply (2)

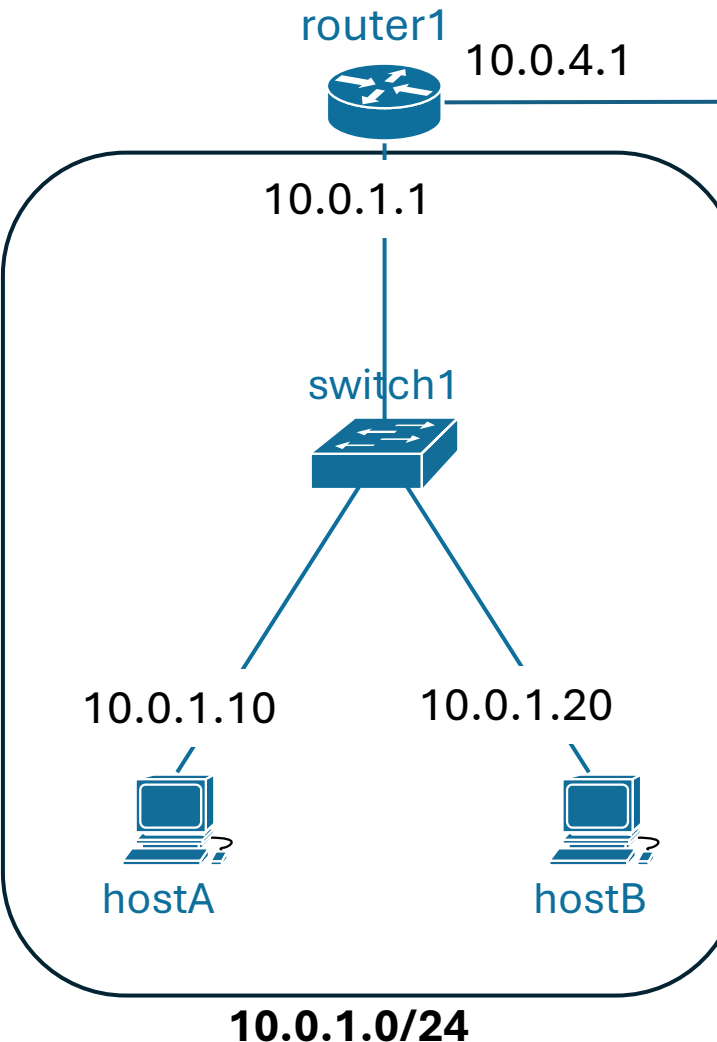
Sender MAC address: 42:3e:e5:36:27:f9 (42:3e:e5:36:27:f9)

Sender IP address: 10.0.1.20

Target MAC address: 7a:57:fd:b3:85:68 (7a:57:fd:b3:85:68)

Target IP address: 10.0.1.10

Arping da Host A a Host B



La tabella ARP è ancora vuota! Infatti, la risposta non fa seguito a una richiesta del sottosistema ARP (infatti, è stata generate da arping)

```
$ ip neigh show dev eth0
```

Se settiamo a 1 il file `/proc/sys/net/ipv4/conf/<interface>/arp_accept` dove **<interface>** è il nome dell'interfaccia, la risposta viene processata:

```
$ sudo su -c "echo 1 > /proc/sys/net/ipv4/conf/eth0/arp_accept"
```

oppure

```
$ sudo sysctl -w net.ipv4.conf.eth0.arp_accept=1
```

Ripete arping e vedere che ora la voce viene creata.

Ciò può introdurre problemi di sicurezza e pertanto è bene ripristinarlo a zero. Si noti che c'è una voce per un certo indirizzo IP nella tabella ARP, qualsiasi risposta che lo hanno come protocol sender viene considerata in ogni caso.

The diagram illustrates a network topology. At the top, a blue router icon is labeled "router1" with the IP address "10.0.4.1" to its right. A blue line connects the router to a blue switch icon labeled "switch1". Above the switch is the IP address "10.0.1.1". The switch is connected to two blue laptop icons. The laptop on the left is labeled "hostA" with the IP address "10.0.1.10" above it. The laptop on the right is labeled "hostB" with the IP address "10.0.1.20" above it. The entire network is enclosed in a rounded rectangle, with the network address "10.0.1.0/24" written in bold black text at the bottom center.

```
$ sudo arping -I eth0 -c 1 -S 0.0.0.0 10.0.1.10
```

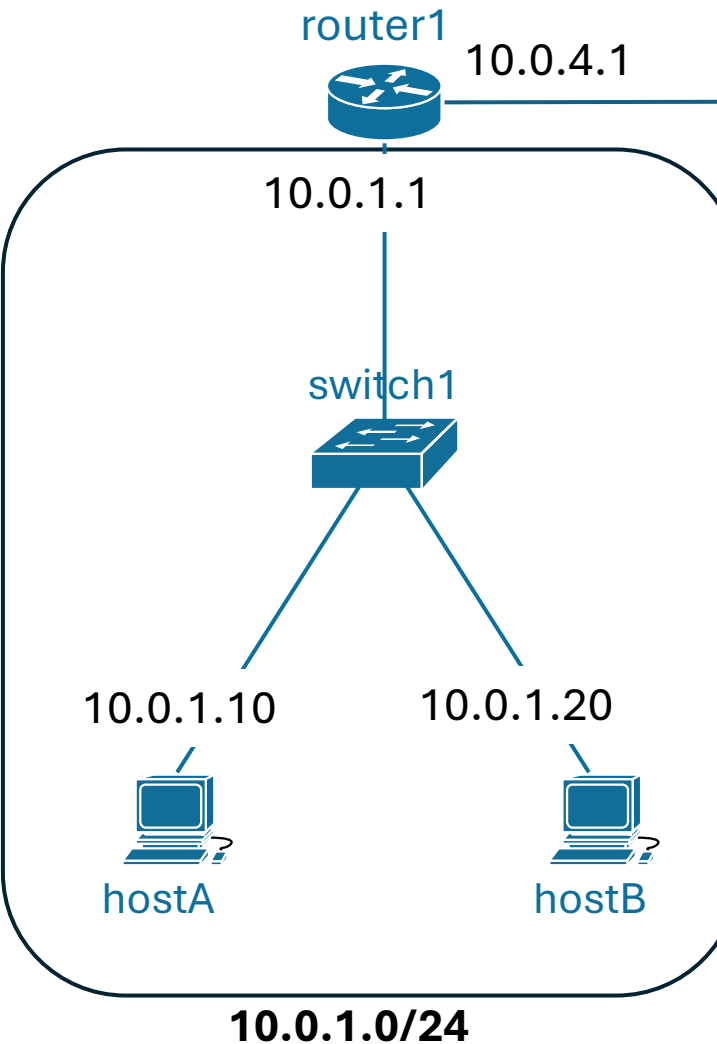
L'opzione `-I` permette di impostare l'interfaccia di uscita.
L'opzione `-S` permette di impostare il sender protocol address a 0.0.0.0.
Per il resto è una richiesta ARP per l'indirizzo 10.0.1.10.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	7a:57:fd:b3:8...	Broadcast	ARP	58	Who has 10.0.1.10? (ARP Probe)

```

    ▶ Frame 1: 58 bytes on wire (464 bits), 58 bytes captured (464 bits) on interface eth0, id 0
    ▼ Ethernet II, Src: 7a:57:fd:b3:85:68 (7a:57:fd:b3:85:68), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
      ▶ Destination: Broadcast (ff:ff:ff:ff:ff:ff)
      ▶ Source: 7a:57:fd:b3:85:68 (7a:57:fd:b3:85:68)
      Type: ARP (0x0806)
      Trailer: 00000000000000000000000000000000
    ▼ Address Resolution Protocol (ARP Probe)
      Hardware type: Ethernet (1)
      Protocol type: IPv4 (0x0800)
      Hardware size: 6
      Protocol size: 4
      Opcode: request (1)
      [Is probe: True]
      Sender MAC address: 7a:57:fd:b3:85:68 (7a:57:fd:b3:85:68)
      Sender IP address: 0.0.0.0
      Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
      Target IP address: 10.0.1.10
  
```

Announcement ARP



L'host A può inviare un Announcement ARP quando inizia a usare l'indirizzo 10.0.1.10 affinché chiunque abbia una voce per quell'indirizzo nella propria tabella ARP, magari per un altro MAC address, aggiorni la corrispondenza usando l'indirizzo MAC dell'host A contenuto nel campo sender protocol address.

```
$ sudo arping -I eth0 -c 1 -A 10.0.1.10
```

L'opzione -I permette di impostare l'interfaccia di uscita, mentre l'opzione -A crea l'announcement per l'indirizzo IP sulla riga di comando.

```
▶ Frame 1: 58 bytes on wire (464 bits), 58 bytes captured (464 bits) on interface eth0, id 0
▼ Ethernet II, Src: 7a:57:fd:b3:85:68 (7a:57:fd:b3:85:68), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  ▶ Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  ▶ Source: 7a:57:fd:b3:85:68 (7a:57:fd:b3:85:68)
  Type: ARP (0x0806)
  Trailer: 00000000000000000000000000000000
▼ Address Resolution Protocol (ARP Announcement)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (1)
  [Is gratuitous: True]
  [Is announcement: True]
  Sender MAC address: 7a:57:fd:b3:85:68 (7a:57:fd:b3:85:68)
  Sender IP address: 10.0.1.10
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 10.0.1.10
```

Announcement ARP

Creiamo una entry "fittizia" nella tabella ARP associata alla scheda eth0 nell'host B:

```
$ sudo ip netns exec hostB ip neigh replace 10.0.1.10 lladdr 11:11:11:11:11:11 nud reachable dev eth0
```

Con il parameter *nud* abbiamo impostato lo stato a reachable, altrimenti sarebbe stato creato come permanent (entry in questo stato non sono cancellate dal comando flush! Occorre un **del** dedicato)

```
$ sudo ip netns exec hostB ip neigh show dev eth0  
10.0.1.10 lladdr 11:11:11:11:11:11 STALE
```

Inviame un announcement dall'Host A:

```
$ sudo ip netns exec hostA arping -I eth0 -c 1 -A 10.0.1.10  
ARPING 10.0.1.10  
Timeout
```

```
--- 10.0.1.10 statistics ---
```

```
1 packets transmitted, 0 packets received, 100% unanswered (0 extra)
```

Verifichiamo che la tabella ARP nell'host B sia stata aggiornata:

```
$ sudo ip netns exec hostB ip neigh show dev eth0  
10.0.1.10 lladdr 7a:57:fd:b3:85:68 STALE
```

rou

10.0.

sw

10.0.1.10/24



hostA

10.0.1.0/24

Announcement ARP

Creiamo una entry "fittizia" nella tabella ARP associata alla scheda eth0 nell'host B:

```
$ sudo ip netns exec hostB ip neigh replace 10.0.1.10 lladdr 11:11:11:11:11:11 nud reachable dev eth0
```

Con il parameter *nud* abbiamo impostato lo stato a reachable, altrimenti sarebbe stato creato come permanent (entry in questo stato non sono cancellate dal comando flush! Occorre un **del** dedicato)

```
$ sudo ip netns exec hostB ip neigh show dev eth0  
10.0.1.10 lladdr 11:11:11:11:11:11 STALE
```

Inviame un announcement dall'Host A:

```
$ sudo ip netns exec hostA arping -I eth0 -c 1 -A 10.0.1.10  
ARPING 10.0.1.10  
Timeout
```

```
--- 10.0.1.10 statistics ---
```

```
1 packets transmitted, 0 packets received, 100% unanswered (0 extra)
```

Verifichiamo che la tabella ARP nell'host B sia stata aggiornata:

```
$ sudo ip netns exec hostB ip neigh show dev eth0  
10.0.1.10 lladdr 7a:57:fd:b3:85:68 STALE
```

rou

10.0.

sw

10.0.1.10/24



hostA

10.0.1.0/24

Riferimenti

- <https://manpages.ubuntu.com/manpages/noble/man8/arping.8.html>