

# Reti informatiche cod. 545II [9 CFU]

Corso di Laurea in Ingegneria Informatica

---

**Laboratorio e Programmazione di Rete**

A. A. 2021/2022

Francesco Pistolesi, PhD

Dipartimento di Ingegneria dell'Informazione

`francesco.pistolesi@unipi.it`

# Lezione 3

## Configurazione di DHCP e test di connettività

# Programma di oggi

- Configurazione di DHCP (server e client)
- Test di connettività
- Configurazione guidata DHCP su Debian

# DHCP

## Dynamic Host Configuration Protocol

# DHCP

*è un protocollo!*

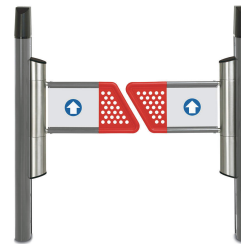
Consente la configurazione **automatica** e **dinamica** dei parametri TCP/IP degli host



Indirizzo  
IP



Maschera  
di rete



Indirizzo  
del gateway



Indirizzo del  
server DNS

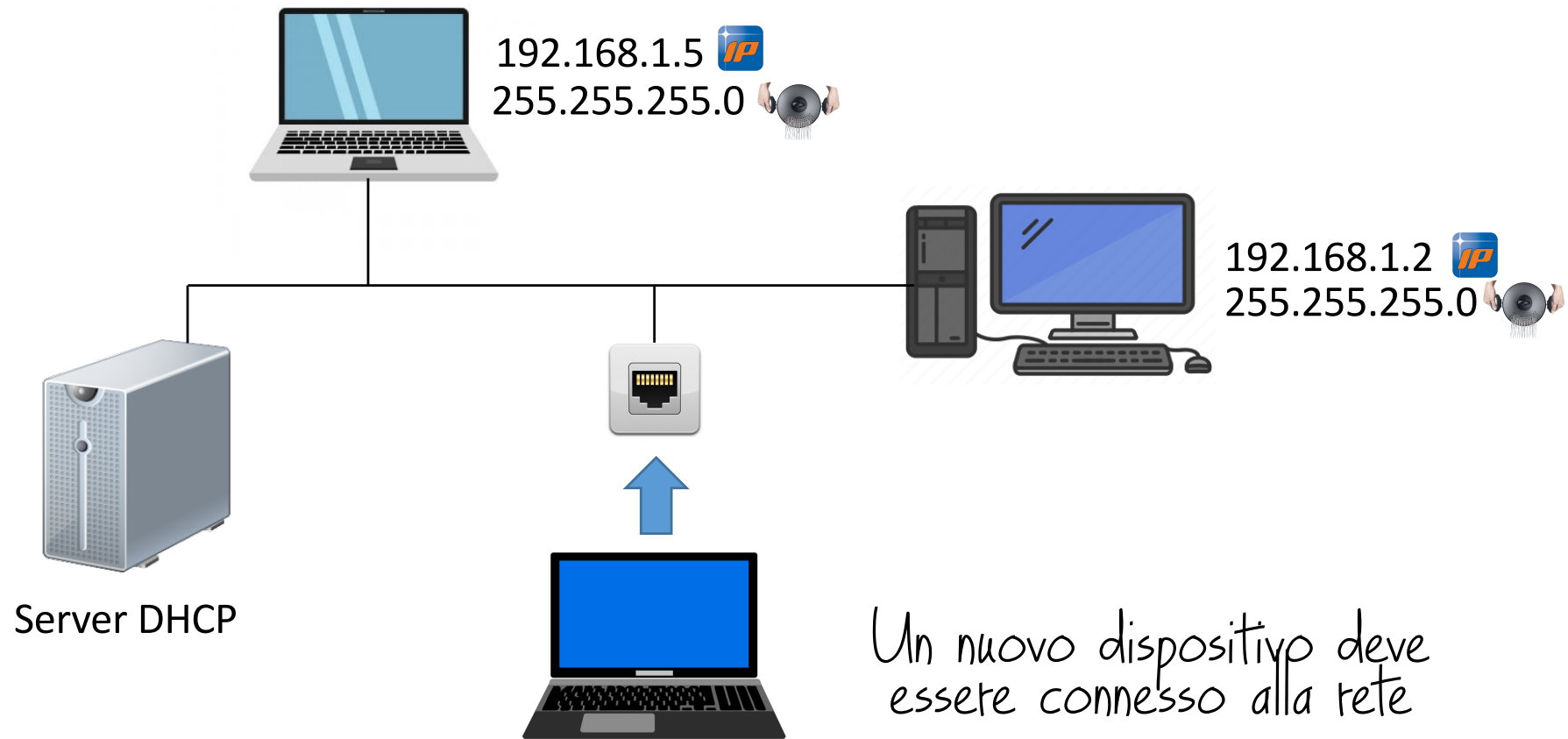
# Reti con DHCP

All'interno della rete c'è un **server DHCP** che configura i parametri di rete degli host



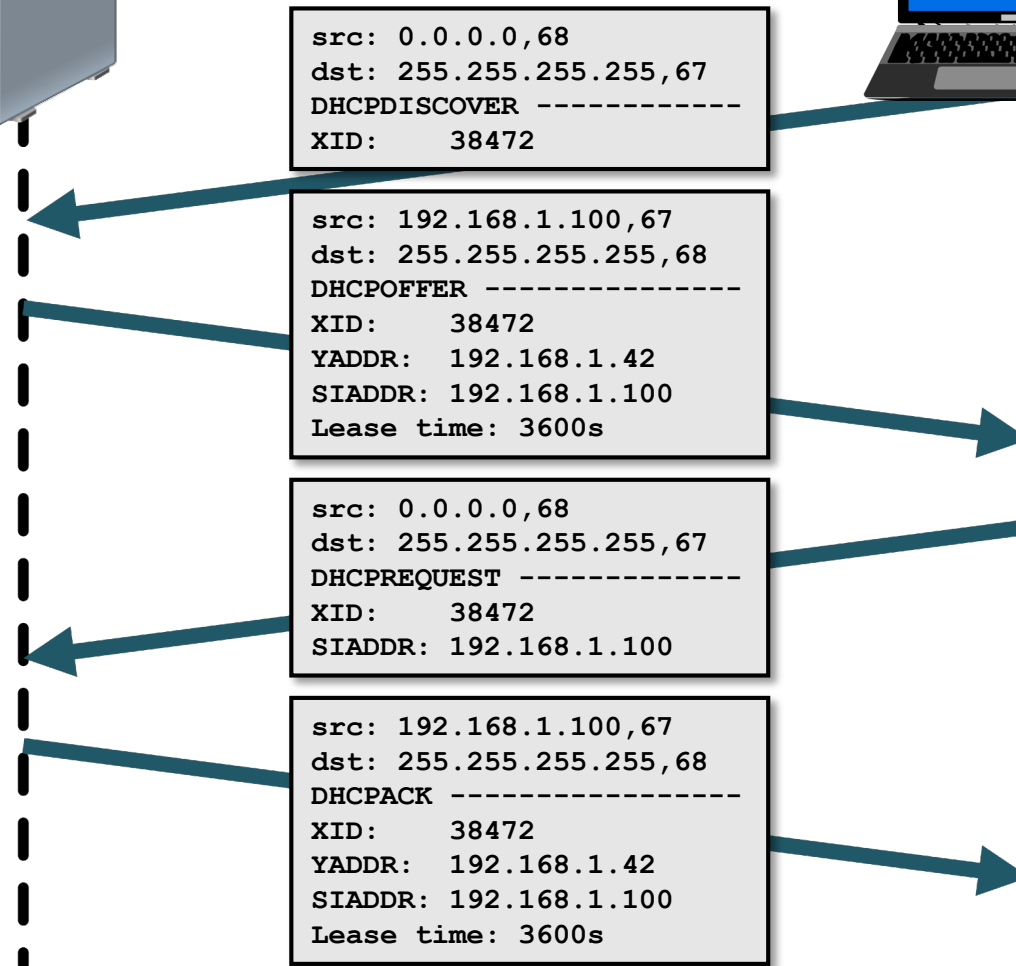
*Non si fa a mano!*

# Come funziona



# Server DHCP

# Host client





# Server DHCP



- Installazione

```
# apt-get install isc-dhcp-server
```

- File di configurazione

- `/etc/default/isc-dhcp-server`

```
...
```

```
INTERFACES="eth0"
```

```
...
```

# Server DHCP



- File di configurazione **`/etc/dhcp/dhcpd.conf`**

```
option domain-name-servers 192.168.0.2, 8.8.8.8;  
option routers 192.168.0.1;  
default-lease-time 3600;  
subnet 192.168.0.0 netmask 255.255.255.0 {  
    range 192.168.0.10 192.168.0.100;  
}
```

**`man dhcpd.conf`**

- Dopo le modifiche:  
    **`# systemctl restart isc-dhcp-server.service`**

# Client DHCP



- File `/etc/network/interfaces`

```
auto lo eth0  
  
iface lo inet loopback  
  
iface eth0 inet dhcp
```

**man interfaces**

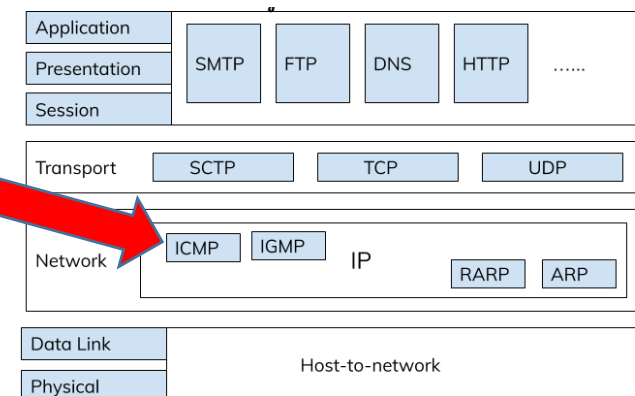
*Invece di 'static' e la configurazione, si mette 'dhcp'*

# Test di connettività

# Internet Control Message Protocol

Protocollo <sup>di servizio</sup> che **rileva malfunzionamenti**, e scambia informazioni di controllo e messaggi di errore

È incapsulato nel protocollo IP

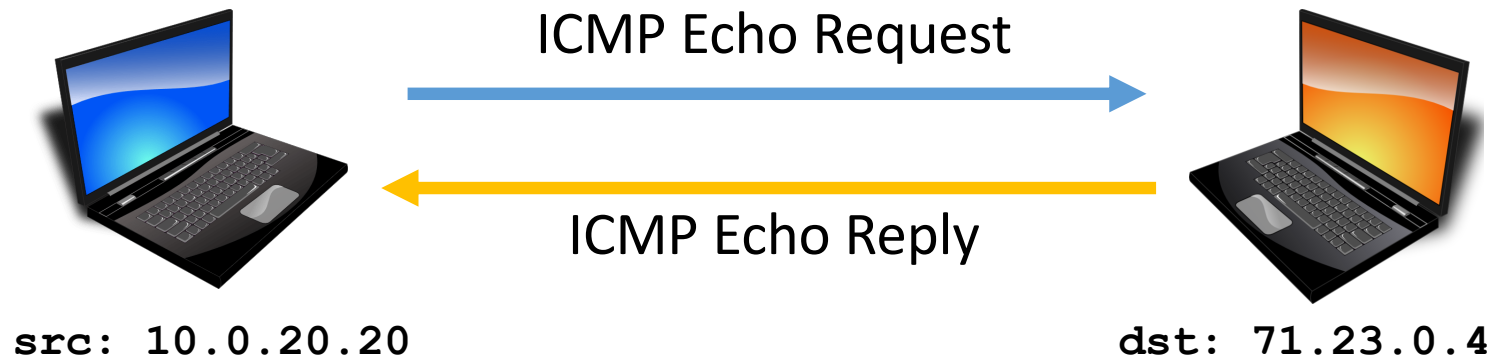


# Comando **ping**

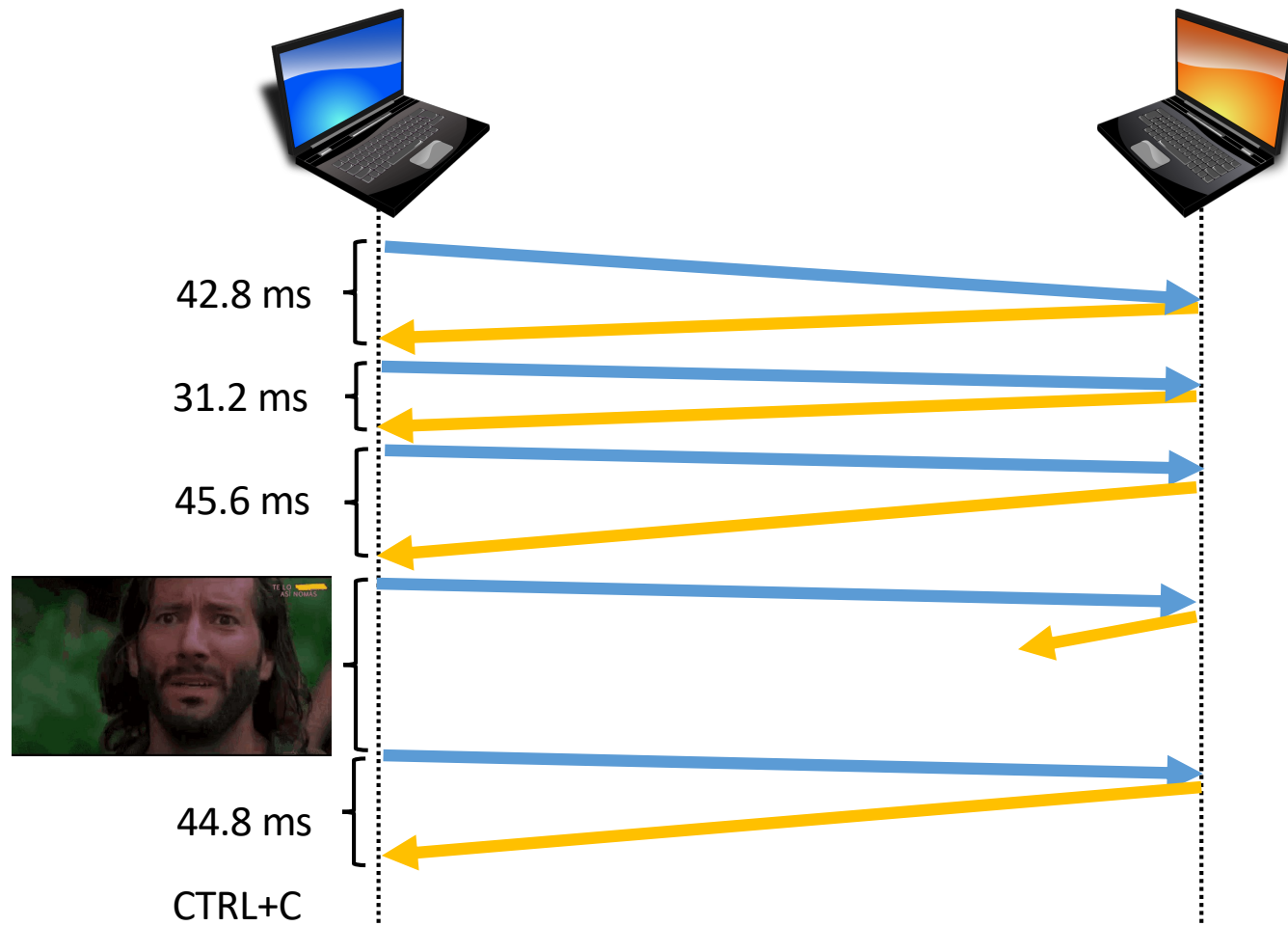
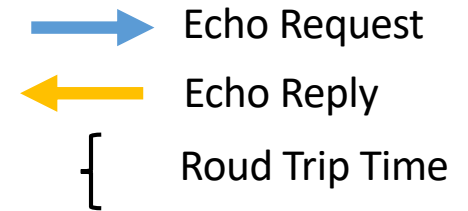
`man ping`

- Testa la connettività tra l'host che lo esegue e un host remoto

```
$ ping www.apple.com
$ ping 192.168.2.34
```
- Iterativamente, invia a un destinatario un messaggio ICMP **Echo Request**, attende un messaggio ICMP **Echo Reply** e misura il tempo (in ms) impiegato a raggiungere l'host destinatario e a ritornare indietro.



# Funzionamento di **ping**



# In dettaglio...

- **ping**, eseguito da A, invia a B una serie di pacchetti *Echo Request* (per default, uno al secondo)
- Quando B riceve un *Echo Request*, invia un pacchetto *Echo Reply* ad A
- **ping** calcola la percentuale di pacchetti ricevuti e il *Round Trip Time (RTT)*
- Al termine del comando, **mostra le statistiche**
  - Se non specificato con le opzioni, termina solo se l'utente lo interrompe con **Ctrl+C**



# Output di **ping**

```
> ping www.google.com
```

```
PING www.google.com (172.217.21.68): 56 data bytes
```

```
64 bytes from 172.217.21.68: icmp_seq=0 ttl=114 time=25.812 ms
```

```
64 bytes from 172.217.21.68: icmp_seq=1 ttl=114 time=25.195 ms
```

```
64 bytes from 172.217.21.68: icmp_seq=2 ttl=114 time=25.446 ms
```

```
64 bytes from 172.217.21.68: icmp_seq=3 ttl=114 time=18.834 ms
```

```
64 bytes from 172.217.21.68: icmp_seq=4 ttl=114 time=26.525 ms
```

```
64 bytes from 172.217.21.68: icmp_seq=5 ttl=114 time=23.773 ms
```

```
64 bytes from 172.217.21.68: icmp_seq=6 ttl=114 time=27.082 ms
```

```
64 bytes from 172.217.21.68: icmp_seq=7 ttl=114 time=25.157 ms
```

```
64 bytes from 172.217.21.68: icmp_seq=8 ttl=114 time=26.405 ms
```

```
64 bytes from 172.217.21.68: icmp_seq=9 ttl=114 time=23.274 ms
```

```
^C
```

```
--- www.google.com ping statistics ---
```

```
10 packets transmitted, 10 packets received, 0.0% packet loss
```

```
round-trip min/avg/max/stddev = 18.834/24.750/27.082/2.268 ms
```

# Errori

Possibili errori:

- *Network unreachable*
  - L'host locale non ha route valide per raggiungere l'host remoto
- *100% packet loss*
  - L'host locale non ha ricevuto alcun pacchetto di risposta
- *Unknown host*
  - Non è stato possibile risolvere il nome di host specificato

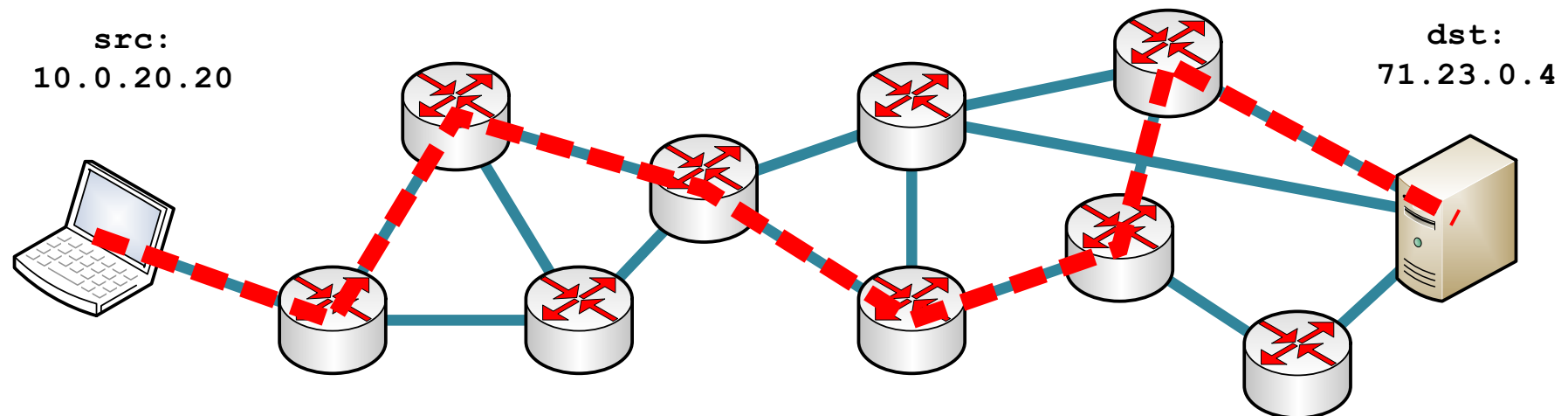
# Opzioni di **ping**

- Opzione **-c** (*count*)
  - Specifica il numero di richieste da inviare
- Opzione **-i** (*interval*)
  - Specifica l'intervallo tra le richieste
- Opzione **-q** (*quiet*)
  - Visualizza solamente le statistiche finali
- Opzione **-s** (*size*)
  - Dimensione in byte del pacchetto, *al netto* degli header ICMP di 8 byte

# Comando **traceroute**

- **Mostra il percorso** che un pacchetto IP effettua per raggiungere un *host destinatario*
- «percorso» = «indirizzi IP dei router attraversati»

```
$ traceroute www.unipi.it
```



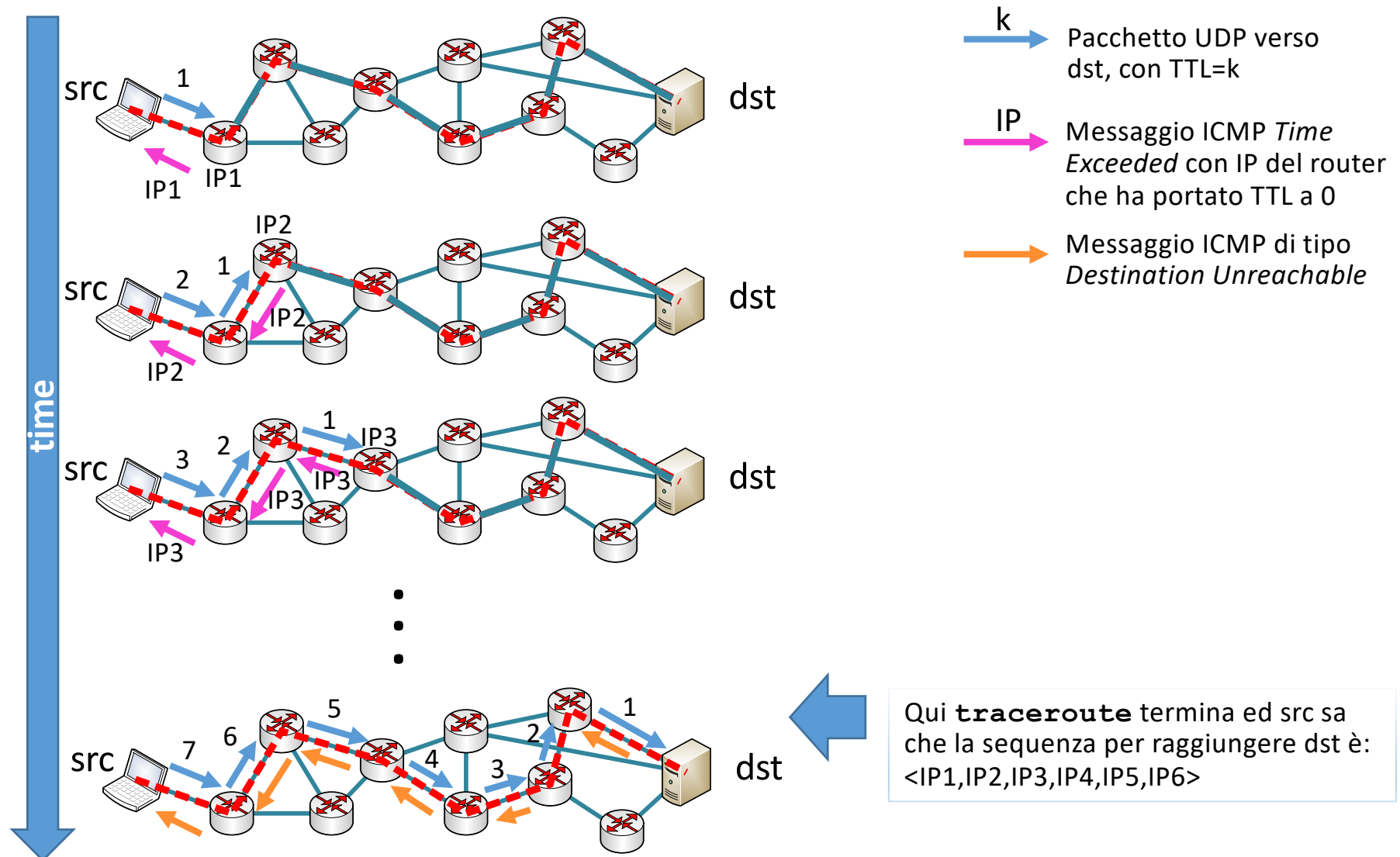
# Gestione di **time-to-live**

- Il **campo TTL** (time-to-leave) del datagram IP specifica il numero di apparati di rete che il pacchetto può attraversare prima di «scadere»
- Ogni router che riceve un pacchetto, prima di inoltrarlo, **diminuisce TTL di 1**. Se TTL=0, invia al mittente un messaggio di errore ICMP di tipo **Time Exceeded** con l'indirizzo del router che ha portato TTL a 0

# Funzionamento di **traceroute**

- **traceroute** sfrutta la gestione di **TTL**
- Invia all'host destinatario una **serie di terne di pacchetti UDP** con **TTL crescente**
  - La prima terna con **TTL=1**, la seconda con **TTL=2**, e così via
- Tiene traccia degli indirizzi IP all'interno dei messaggi **ICMP Time Exceeded** ricevuti finché l'ultimo pacchetto della serie raggiunge l'host destinatario
- Il destinatario, quando viene raggiunto, non manda ICMP Time Exceeded, ma un **ICMP Destination Unreachable**
  - UDP ha bisogno di una porta. I pacchetti verso il destinatario necessitano di una porta in ascolto. Scegliendone una alla cieca, è improbabile beccarne una in ascolto, ecco perché l'host invia Destination Unreachable (con *code=3*, cioè *port unreachable*).
- **traceroute** ordina gli IP in base al TTL crescente e ricostruisce il percorso. Fornisce anche i round-trip dei pacchetti

# Per capire...



# Output di **tracert**

Round-trip time prima terna (TTL=1)

IP del primo hop

```
> traceroute www.google.com
traceroute to www.google.com (216.58.205.196), 30 hops max, 60 byte packets
 1  131.114.175.73 (131.114.175.73)  0.348 ms  0.351 ms  0.378 ms
 2  jser-jing.unipi.it (131.114.191.129)  0.363 ms  0.371 ms  0.398 ms
 3  ru-unipi-rx1-pil.pil.garr.net (193.206.136.13)  1.302 ms  1.312 ms  1.321 ms
 4  rx1-pil-rx2-mi2.mi2.garr.net (90.147.80.210)  6.881 ms  6.899 ms  6.898 ms
 5  72.14.214.105 (72.14.214.105)  7.023 ms  6.974 ms  6.986 ms
 6  * * *
 7  216.239.42.27 (216.239.42.27)  7.020 ms  6.931 ms  6.997 ms
 8  mil04s29-in-f4.1e100.net (216.58.205.196)  6.949 ms  7.090 ms  7.087 ms
```

Hop identifiers

**Scaduto il timeout** (default 5 sec): non è stato ricevuto il messaggio di risposta per alcun pacchetto della terna. Possibili cause:

- device non configurato per rispondere a traffico ICMP/UDP;
- pacchetti scartati per motivi di rete (es: traffico bloccato da un firewall).



# Svantaggi

1. I pacchetti possono seguire più percorsi, quindi gli **IP ricavati possono riferirsi a più percorsi**
2. Se i pacchetti e i messaggi ICMP seguono percorsi diversi, il **calcolo del round-trip time è inaffidabile.**

