

Strumenti di analisi della rete

*Valentina Ceoletta, Marco De Bona, Federico De Meo,
Luigi Capogrosso e Davide Quaglia*

1. Introduzione agli analizzatori di rete

Esistono diversi strumenti SW che consentono di analizzare i pacchetti che arrivano alla propria interfaccia di rete:

- **TCPDUMP** [[qui](#)], storico tool da linea di comando (per OS Linux);
- **WinDump** [[qui](#)], storico tool da linea di comando (per OS Windows);
- **Wireshark** [[qui](#)], moderno tool con GUI disponibile per Linux, Windows e Mac.

Tutti i precedenti SW elencati si basano sulla libreria C **libpcap** [[qui](#)]. Le principali funzionalità di questa libreria sono:

- Possibilità di cercare e trovare interfacce di rete;
- Gestione avanzata di filtri di cattura;
- Gestione degli errori e statistiche di cattura.

1.1 Download ed installazione

I tool possono essere scaricati liberamente dalle rispettive pagine web indicate, oppure, possono essere già presenti nell'installazione della propria distribuzione.

ATTENZIONE: per poter utilizzare le funzionalità di cattura diretta in ambiente Linux bisogna essere autenticati come utente `root`, oppure, aver installato il tool con `setuid a root`.

1.2 Sniffing: Concetti chiave

- **Sniffing all'interno di reti non-switched.** In questa tipologia di reti il mezzo trasmissivo è condiviso e, quindi, tutte le schede di rete dei PC ricevono tutti i pacchetti, anche quelli destinati ad altri. I propri, invece, sono selezionati a seconda dell'indirizzo MAC (indirizzo hardware specifico della scheda di rete).

Lo sniffing, in questo caso, consiste nell'impostare sull'interfaccia di rete la cosiddetta **modalità promiscua** che disattiva il “filtro hardware” basato sul MAC. Così facendo, si permette al sistema l'ascolto di tutto il traffico passante sul cavo. Un esempio di rete non-switched è la rete **WiFi**.

- **Sniffing all'interno di reti Ethernet switched.** In questo caso, invece, l'apparato centrale della rete (definito switch), si preoccupa di inoltrare su ciascuna porta solo il traffico destinato ai dispositivi collegati a quella porta. Quindi, ciascuna interfaccia di rete, riceve solo i pacchetti destinati al proprio indirizzo, i pacchetti multicast e quelli broadcast.

L'impostazione della modalità promiscua è, pertanto, insufficiente per poter intercettare il traffico in una rete gestita da switch.

2. Utilizzo di Wireshark

Alcune caratteristiche:

- I dati possono essere acquisiti direttamente dall'interfaccia di rete (reti Ethernet, WiFi, ADSL, ecc...), oppure, possono essere letti su un file di cattura precedente;
- I dati di rete catturati possono essere esplorati nelle loro parti tramite un'interfaccia grafica;
- I filtri di visualizzazione possono essere usati per colorare o visualizzare le informazioni sommarie sui pacchetti;
- I protocolli di comunicazione possono essere scomposti dato che Wireshark riesce a “comprendere” la struttura dei diversi protocolli di rete. È quindi possibile visualizzare incapsulamenti, campi singoli e interpretare il loro significato;
- È possibile studiare le statistiche di una connessione TCP e di estrarne il contenuto.

Per eseguire l'applicazione, digitare da terminale:

```
$ wireshark
```

oppure, cliccando sull'apposita icona:



2.1 Cattura dei pacchetti

Per avviare la cattura dei pacchetti è necessario specificare da quale interfaccia si vuole effettuare la cattura. Per fare ciò, aprire il menu **Capture/Interfaces** e, la nuova finestra, chiederà quale interfaccia di rete utilizzare per la cattura (*Figura 1*).

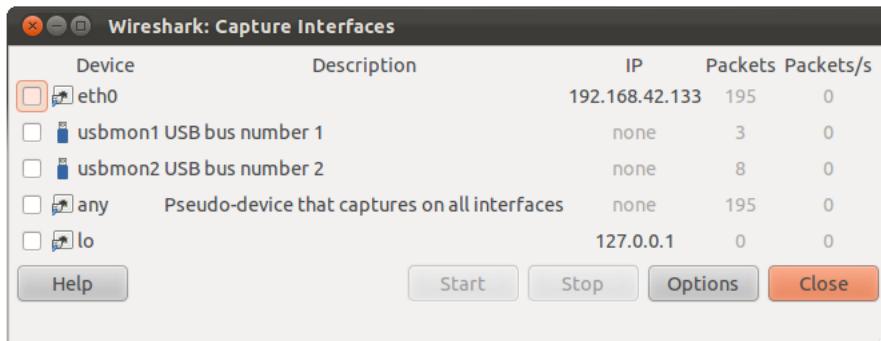


Figura 1 Interfacce di cattura.

Solitamente, in Linux, la prima interfaccia di rete ha come nome **eth0**, mentre, in Windows, ha il nome del produttore della scheda.

Una volta individuata l'interfaccia, premendo il tasto **Options** è possibile applicare diverse impostazioni, tra le quali, anche i filtri da utilizzare per la cattura (*Figura 2*).

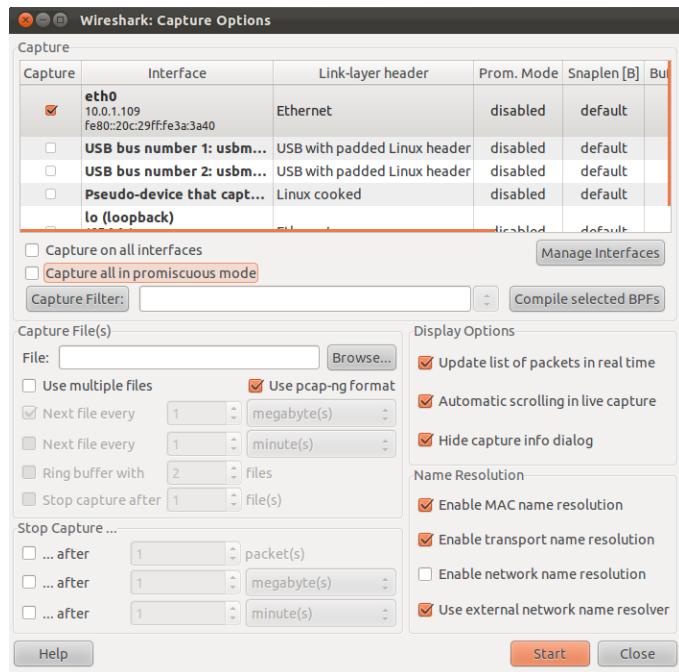


Figura 2 Opzioni di cattura.

Premendo **Start** si avvia la cattura. Per terminare la cattura (se non si è impostato un limite in pacchetti, secondi o byte) si usa il comando **Stop** dal menu **Capture**.

ATTENZIONE: Sui sistemi Linux, i nomi delle interfacce di rete potrebbero essere nel nuovo formato dove, **eth** è sostituito da **enp** e **wlan** è sostituito da **wlp**, ecc...

2.2 Applicazione dei filtri nella cattura

È possibile limitare la cattura ai soli pacchetti che rispettano specifici requisiti impostando un filtro di cattura nella finestra **Capture Options** e premendo sul tasto **Capture Filter** (*Figura 3*).

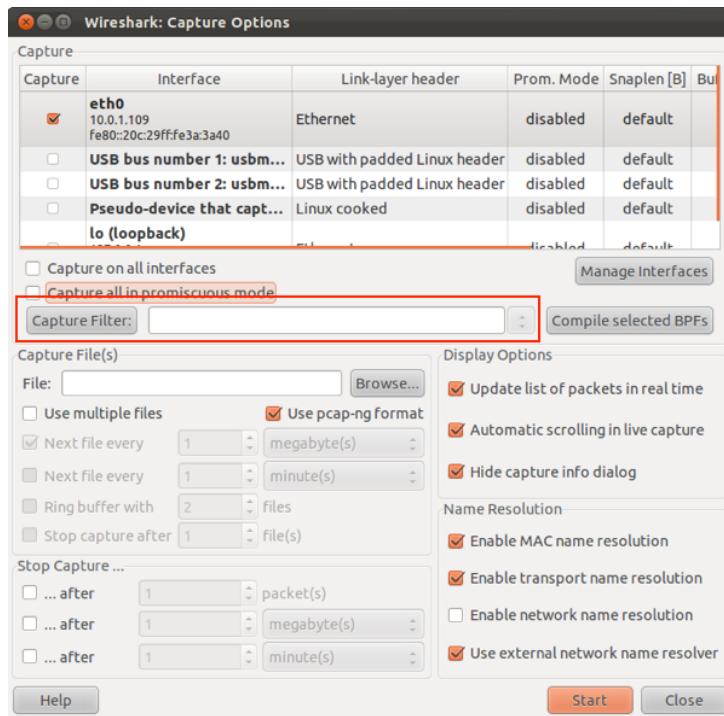


Figura 3.

Quindi, è possibile scrivere l'espressione del filtro manualmente (trovate la sintassi [[qui](#)]), oppure, selezionarne uno preesistente cliccando sull'apposito bottone e, al limite, modificarlo (*Figura 4*).



Figura 4 Filtri di cattura già pronti.

I filtri di cattura programmano la scheda di rete con l'intento di catturare solo determinati pacchetti. Questi, sono solitamente utilizzati quando la quantità di pacchetti che passano sul tratto di rete osservato è tale per cui, se tutti i pacchetti venissero passati alla CPU, le sue prestazioni sarebbero compromesse.

Al termine della cattura la finestra principale di Wireshark mostra i dati catturati.

2.3 Finestra principale

La finestra è divisa in tre parti, nell'ordine:

1. Tabella dei pacchetti catturati;
 2. Vista sulla encapsulazione dei protocolli del pacchetto selezionato nella tabella;
 3. Vista in versione binaria dei dati del pacchetto selezionato nella tabella.

The screenshot shows a Wireshark capture of network traffic. The packet list pane displays 12 captured frames, with frame 6 highlighted in blue. The details pane shows the following information for frame 6:

- Frame 6: 434 bytes on wire (3472 bits), 434 bytes captured (3472 bits)**
- Ethernet II, Src: Dell_a7:a0:a8 (00:24:e8:a7:a0:a8), Dst: Cisco_7b:b7:cb (00:1f:9d:7b:b7:cb)**
- Internet Protocol Version 4, Src: 157.27.252.202 (157.27.252.202), Dst: 130.192.73.1 (130.192.73.1)**
- Transmission Control Protocol, Src Port: 36986 (36986), Dst Port: http (80), Seq: 1, Ack: 1, Len: 380**
- Hypertext Transfer Protocol**

The packet details pane shows the raw hex and ASCII data for frame 6. The bytes pane shows the raw data in a hex dump format.

Inoltre, è possibile visualizzare un sommario scegliendo il menu **Statistics/Summary**.

2.4 Regole di colorazione

È possibile migliorare la visualizzazione dei vari pacchetti nella tabella principale colorando le righe in base al tipo di protocollo, oppure, di indirizzi coinvolti. I filtri di coloramento possono essere impostati nel menu **View/Colouring rules**.

È possibile creare nuovi filtri di colori attraverso il pulsante **New** (*Figura 5*) che apre una finestra in cui è possibile impostare nome, regole e colori.

Le regole sono espressioni booleane sui campi del pacchetto; si impostano con un linguaggio diverso da quello dei filtri di cattura (si noti il pulsante **Expression** che semplifica la scrittura delle regole combinando campi dei pacchetti, valori e operatori logici).

Per ciascun pacchetto da visualizzare, le regole di colorazione sono considerate dal programma “dall’alto verso il basso”. Quando una regola è soddisfatta, il pacchetto viene visualizzato con i colori corrispondenti. Se tutte le regole vengono passate in rassegna e nessuna è vera, il pacchetto viene visualizzato in nero su bianco.

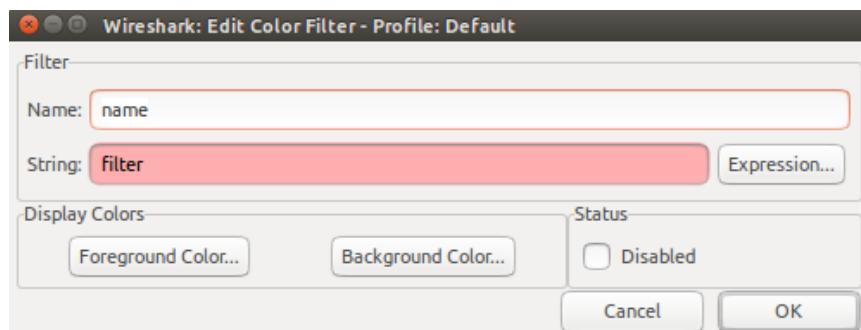
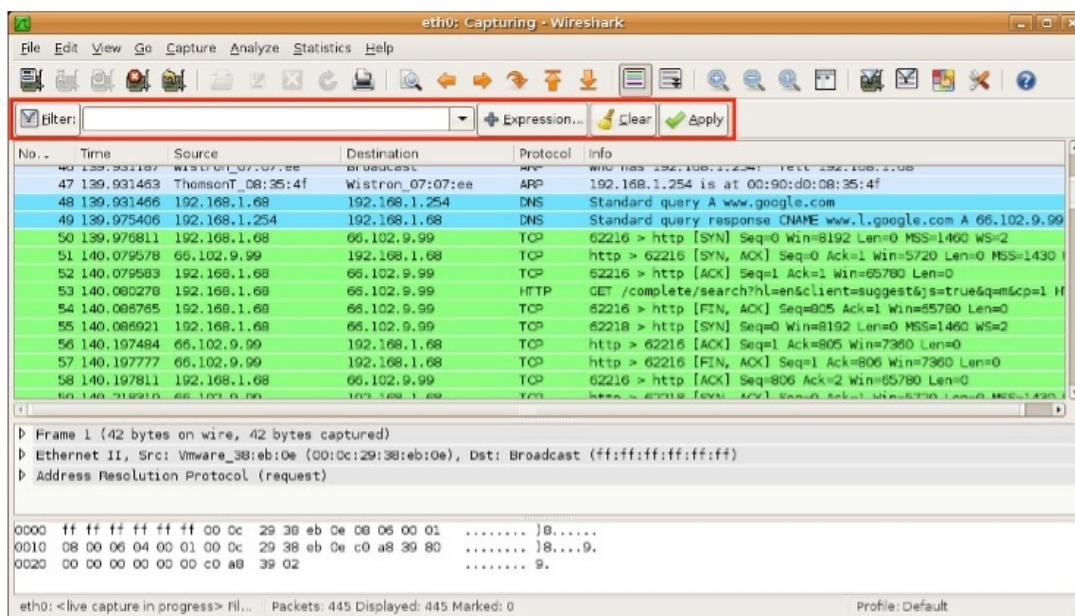


Figura 5 Creazione di una nuova regola di colorazione.

2.5 Applicazione dei filtri di visualizzazione nella finestra principale

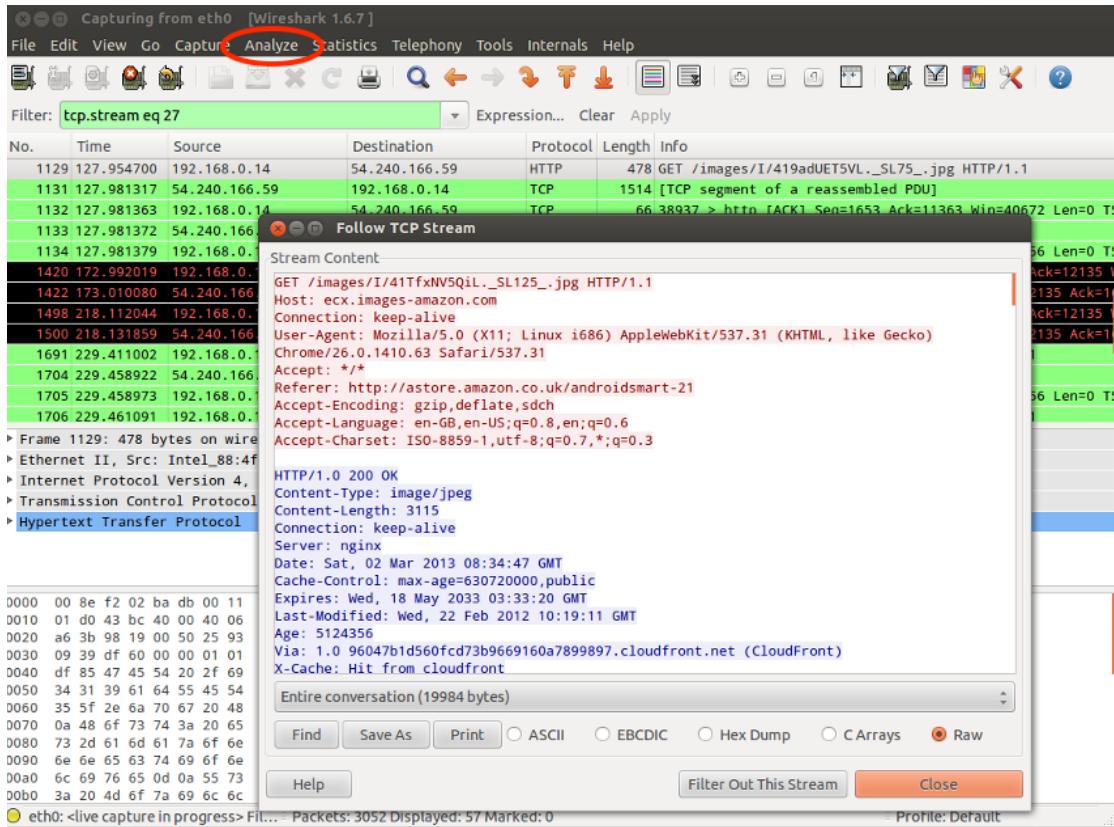
È possibile creare un filtro per limitare il numero di pacchetti visualizzati in una cattura già avvenuta. Per fare questo, si utilizza la barra dei filtri presente nella schermata principale.



Attraverso il tasto **Filter** è possibile specificare un filtro esistente, oppure, crearne di nuovi. Per la stesura dei filtri ci aiuta il tasto **Expression** che fornisce un tool automatico di stesura filtri simile a quello per la colorazione delle righe.

2.6 Analisi del flusso TCP

Per quanto riguarda TCP, selezionando un pacchetto TCP nella finestra principale, è possibile seguire l'intero flusso dati di quella “conversazione” mediante la voce **Analyze/Follow TCP Stream** e, quindi, studiare l'andamento di alcuni parametri del protocollo mediante la voce **Statistics/TCP Stream Content**.



2.7 Visualizzazione del livello Data-link delle PDU non Ethernet

Mentre il driver della scheda Ethernet fornisce a Wireshark l'esatto header della PDU di livello Data-link, alcune altre interfacce di rete non-Ethernet (es. WiFi) potrebbero non farlo a meno dell'utilizzo di plugin specifici per Wireshark (o per la scheda di rete). Nel caso in cui tale header non venga fornito, Wireshark visualizzerà un header Ethernet “autostruito”, oppure, un header con la dicitura **linux cooked capture**.

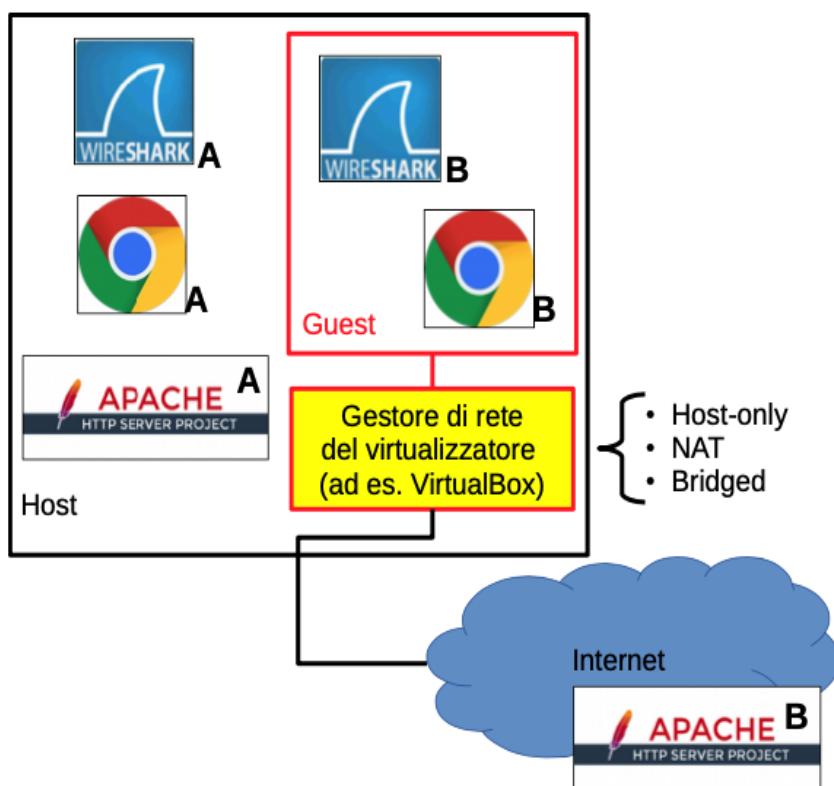
Per il caso specifico delle interfacce WiFi si veda come approfondimento: [WLAN \(IEEE 802.11\) capture setup](#).

2.8 Cattura di traffico di rete all'interno di una macchina virtuale

Particolarmente interessante è la cattura del traffico di rete all'interno di una macchina virtuale. La figura seguente mostra un sistema Host che ospita un sistema Guest attraverso il meccanismo della virtualizzazione.

Il traffico di rete catturabile dall'istanza B di Wireshark nel sistema Guest dipende dalla configurazione del gestore di rete del virtualizzatore che prevede le seguenti 3 alternative principali:

- **“Host-only”**. L'unico traffico di rete possibile (e quindi catturabile) nel Guest è quello tra Guest e Host (ad esempio l'istanza B del browser comunica con l'istanza A di HTTP server sull'Host);
- **“NAT”**. L'unico traffico di rete nel Guest è quello verso l'esterno (ad esempio tra l'istanza B del browser e l'istanza B di HTTP server su Internet);
- **“Bridged”**. Il gestore di rete del virtualizzatore simula il comportamento di uno switch Ethernet a cui sono collegati sia l'Host sia il Guest. In questo scenario, l'istanza B di Wireshark può catturare sia il traffico legato al Guest sia quello legato all'Host di tipo broadcast.



3. Comando ping

```
luigi@capogrosso ~ % ping
usage: ping [-AaDdfnoQqRrv] [-c count] [-G sweepmaxsize]
            [-g sweepminsize] [-h sweepincrsize] [-i wait]
            [-l preload] [-M mask | time] [-m ttl] [-p pattern]
            [-S src_addr] [-s packetsize] [-t timeout][ -W waittime]
            [-z tos] host
ping [-AaDdfLnoQqRrv] [-c count] [-I iface] [-i wait]
      [-l preload] [-M mask | time] [-m ttl] [-p pattern] [-s
src_addr] [-s packetsize] [-T ttl] [-t timeout] [-W waittime]
      [-z tos] mcast-group
```

Il comando **ping** è un semplice strumento per verificare la raggiungibilità di un computer connesso alla rete e il relativo **Round Trip Time (RTT)**, ossia il tempo che intercorre dalla partenza del pacchetto inviato al ritorno della risposta. Per questa operazione viene utilizzato il protocollo **ICMP**.

ICMP (Internet Control Message Protocol) è un protocollo di servizio per trasmettere informazioni riguardanti malfunzionamenti (es. **TimeExceeded** oppure **Destination Unreachable**), informazioni di controllo (es. **Echo Request** oppure **Echo Replay**) o messaggi tra i vari componenti di una rete di calcolatori.

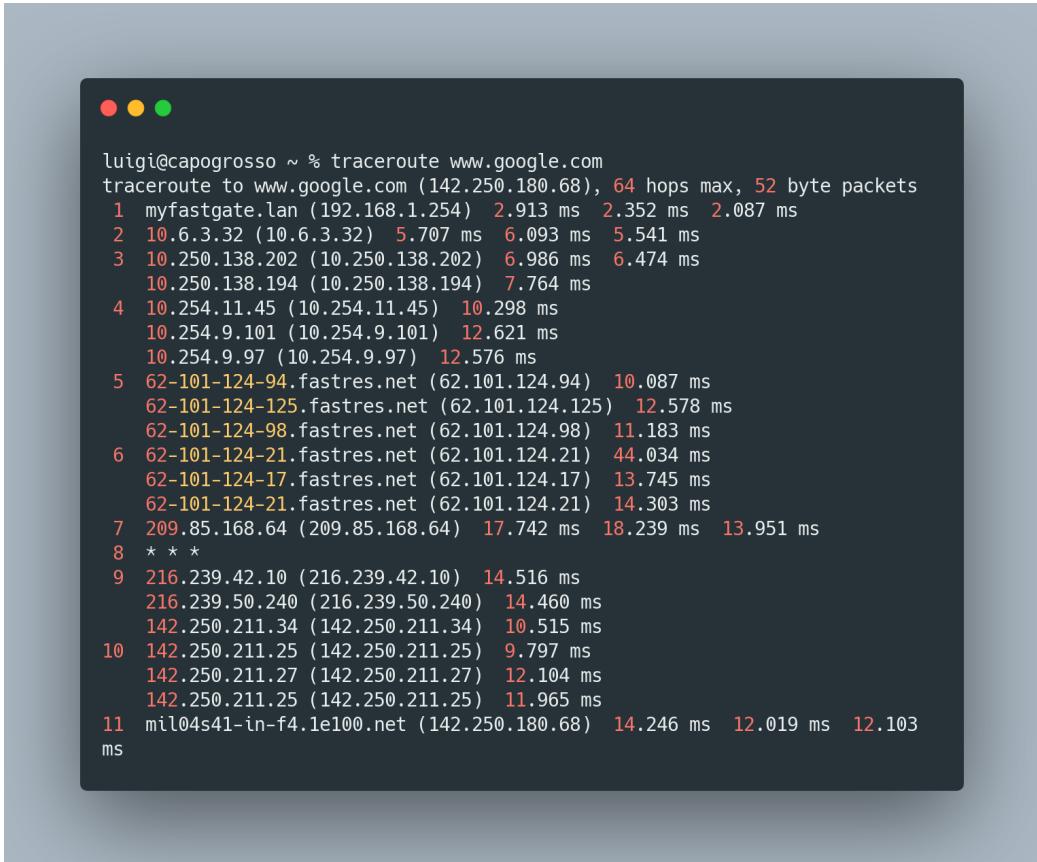
```
luigi@capogrosso ~ % ping www.google.com
PING www.google.com (142.250.180.132): 56 data bytes
64 bytes from 142.250.180.132: icmp_seq=0 ttl=118 time=9.748 ms
64 bytes from 142.250.180.132: icmp_seq=1 ttl=118 time=8.970 ms
64 bytes from 142.250.180.132: icmp_seq=2 ttl=118 time=10.094 ms
64 bytes from 142.250.180.132: icmp_seq=3 ttl=118 time=9.784 ms
64 bytes from 142.250.180.132: icmp_seq=4 ttl=118 time=10.611 ms
64 bytes from 142.250.180.132: icmp_seq=5 ttl=118 time=10.133 ms
64 bytes from 142.250.180.132: icmp_seq=6 ttl=118 time=9.829 ms
64 bytes from 142.250.180.132: icmp_seq=7 ttl=118 time=8.965 ms
^C
--- www.google.com ping statistics ---
8 packets transmitted, 8 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 8.965/9.767/10.611/0.529 ms
```

Figura 6 Esempio di esecuzione del comando **ping**.

Dopo aver avviato il comando **ping** da terminale è possibile visualizzare lo scambio dei messaggi tra il nostro calcolatore e la destinazione. Per fare ciò è sufficiente avviare una cattura da Wireshark applicando il filtro **icmp** per la visualizzazione.

4. Comando traceroute

Il comando **traceroute** (**tracert** in Windows) è, invece, un semplice strumento per tracciare il percorso che un pacchetto segue dalla sorgente alla destinazione. Il comando mostra un elenco di tutte le interfacce dei router che il pacchetto attraversa finché non raggiunge la destinazione.



```
luigi@capogrosso ~ % traceroute www.google.com
traceroute to www.google.com (142.250.180.68), 64 hops max, 52 byte packets
 1 myfastgate.lan (192.168.1.254) 2.913 ms 2.352 ms 2.087 ms
 2 10.6.3.32 (10.6.3.32) 5.707 ms 6.093 ms 5.541 ms
 3 10.250.138.202 (10.250.138.202) 6.986 ms 6.474 ms
 10.250.138.194 (10.250.138.194) 7.764 ms
 4 10.254.11.45 (10.254.11.45) 10.298 ms
 10.254.9.101 (10.254.9.101) 12.621 ms
 10.254.9.97 (10.254.9.97) 12.576 ms
 5 62-101-124-94.fastres.net (62.101.124.94) 10.087 ms
 62-101-124-125.fastres.net (62.101.124.125) 12.578 ms
 62-101-124-98.fastres.net (62.101.124.98) 11.183 ms
 6 62-101-124-21.fastres.net (62.101.124.21) 44.034 ms
 62-101-124-17.fastres.net (62.101.124.17) 13.745 ms
 62-101-124-21.fastres.net (62.101.124.21) 14.303 ms
 7 209.85.168.64 (209.85.168.64) 17.742 ms 18.239 ms 13.951 ms
 8 * * *
 9 216.239.42.10 (216.239.42.10) 14.516 ms
 216.239.50.240 (216.239.50.240) 14.460 ms
 142.250.211.34 (142.250.211.34) 10.515 ms
 10 142.250.211.25 (142.250.211.25) 9.797 ms
 142.250.211.27 (142.250.211.27) 12.104 ms
 142.250.211.25 (142.250.211.25) 11.965 ms
 11 mil04s41-in-f4.1e100.net (142.250.180.68) 14.246 ms 12.019 ms 12.103
ms
```

Figura 8 Esempio di output del comando Traceroute.

Si noti la presenza di alcuni asterischi in corrispondenza di determinate tappe. Questi sono dovuti al fatto che, certe interfacce di specifici router, non forniscono alcuna informazione. Questa scelta viene presa dagli amministratori di rete per evitare di svelare la topologia di rete a possibili malware. In tal caso **traceroute** non può mostrare tali passi del percorso.

Dopo aver avviato il comando **traceroute** da terminale è possibile visualizzare in Wireshark lo scambio dei messaggi tra il nostro calcolatore e la sorgente di destinazione. Per fare ciò è sufficiente avviare una cattura da Wireshark applicando il filtro **icmp** per la visualizzazione.

5. Comando nslookup

Il comando **nslookup** consente di effettuare una interrogazione ai server **DNS** per poter ottenere da un hostname il relativo indirizzo IP, o viceversa. Si può utilizzare in due modalità:

- Interattivo;
- Non interattivo.

DNS (Domain Name System) è un sistema di server organizzato gerarchicamente, per la gestione del namespace (Domain Name Space). Il compito principale di questo servizio è quello di rispondere alle richieste della risoluzione del nome di dominio, ovvero la conversione dei nomi di dominio in indirizzi IP.

Modalità interattiva. Permette di effettuare più interrogazioni e visualizza i singoli risultati. Viene abilitata in maniera automatica quando il comando non è seguito da alcun argomento.

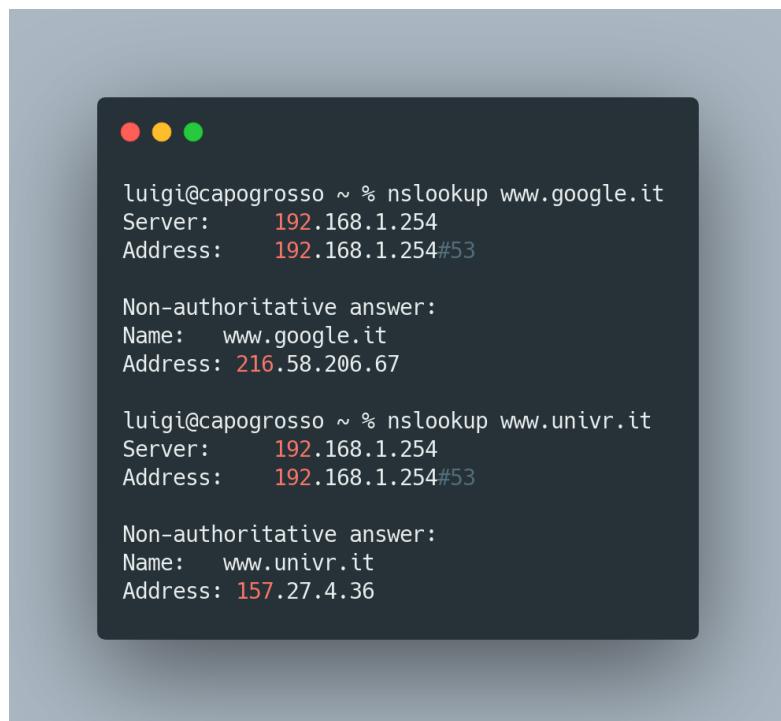


```
luigi@capogrosso ~ % nslookup
> www.google.com
Server:  192.168.1.254
Address: 192.168.1.254#53

Non-authoritative answer:
Name: www.google.com
Address: 216.58.208.132
> www.univr.it
Server: 192.168.1.254
Address: 192.168.1.254#53

Non-authoritative answer:
www.univr.it canonical name = aol-prod.univr.it.
Name: aol-prod.univr.it
Address: 157.27.4.36
>
```

Modalità non interattiva. Permette di effettuare una sola interrogazione visualizzandone il risultato. Abilitata ogni qualvolta si specifichi l'host-to-find.



```
luigi@capogrosso ~ % nslookup www.google.it
Server: 192.168.1.254
Address: 192.168.1.254#53

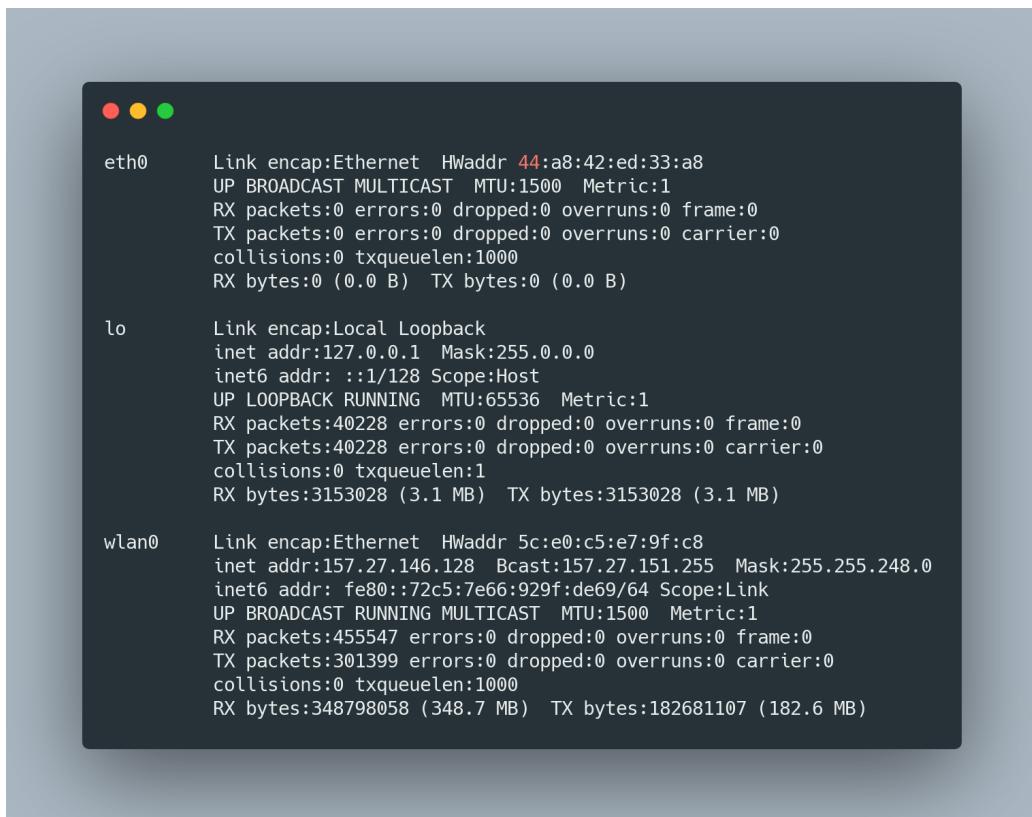
Non-authoritative answer:
Name: www.google.it
Address: 216.58.206.67

luigi@capogrosso ~ % nslookup www.univr.it
Server: 192.168.1.254
Address: 192.168.1.254#53

Non-authoritative answer:
Name: www.univr.it
Address: 157.27.4.36
```

6. Comando ifconfig

Il comando **ifconfig** (ipconfig in Windows) è un utilizzato per configurare e controllare un'interfaccia di rete TCP/IP da riga di comando. L'esecuzione del comando con l'opzione **-a** mostra a video le informazioni di tutte le interfacce di rete.

A screenshot of a terminal window on a Mac OS X desktop. The window has red, yellow, and green status icons at the top. The terminal displays the output of the 'ifconfig -a' command. The output shows three network interfaces: eth0 (Ethernet), lo (loopback), and wlan0 (wireless). Each interface lists its link layer information (HWaddr), state (UP/BROADCAST/MULTICAST), MTU, Metric, and various statistics (RX/TX packets, errors, dropped, overruns, collisions, bytes).

- **eth0** è la prima interfaccia Ethernet (ulteriori interfacce, se presenti, saranno denominate **eth1, eth2, etc...**)
- **lo** è l'interfaccia loopback, sempre presente. È un'interfaccia di rete “speciale” che il sistema utilizza per comunicare con sé stesso.
- **wlan0** è il nome della prima interfaccia di rete wireless del sistema. Ulteriori interfacce wireless saranno denominate **wlan1, wlan2, etc...**

Quella precedentemente mostrata è la vecchia convenzione dei nomi per le interfacce di rete all'interno del sistema Linux (altri OS potrebbero avere differenti convenzioni).

La nuova nomenclatura è, invece, quella qui di seguito mostrata:

```
● ● ●

enp2s0    Link encap:Ethernet HWaddr 44:a8:42:ed:33:a8
          UP BROADCAST MULTICAST MTU:1500 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

lo       Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:65536 Metric:1
          RX packets:51406 errors:0 dropped:0 overruns:0 frame:0
          TX packets:51406 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:5494472 (5.4 MB) TX bytes:5494472 (5.4 MB)

wlp3s0   Link encap:Ethernet HWaddr 5c:e0:c5:e7:9f:c8
          inet addr:192.168.43.173 Bcast:192.168.43.255 Mask:255.255.255.0
          inet6 addr: fe80::2a4b:3d6b:685:1687/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:922741 errors:0 dropped:0 overruns:0 frame:0
          TX packets:482874 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:725571328 (725.5 MB) TX bytes:161246826 (161.2 MB)
```

7. Comando route

Il comando **route** (**route PRINT** su Windows) è utilizzato per visualizzare e modificare le tabelle di routing. L'esecuzione permette di visualizzare la tabella di routing dell'host come nell'esempio seguente:

```
● ● ●

Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref  Use Iface
default         192.168.43.1   0.0.0.0        UG    600    0      0 wlp3s0
link-local      *              255.255.0.0    U     1000   0      0 wlp3s0
192.168.43.0   *              255.255.255.0  U     600    0      0 wlp3s0
192.168.122.0  *              255.255.255.0  U     0      0      0 virbr0
```

8. Comando whois

Il comando **whois** consente, mediante l'interrogazione di appositi database server da parte di un client, di stabilire il nome del privato, azienda o ente al quale è intestato un determinato indirizzo IP o uno specifico dominio DNS. Nel Whois vengono solitamente mostrate anche informazioni riguardanti l'intestatario, data di registrazione e la data di scadenza.
Whois si può consultare tradizionalmente da riga di comando, anche se ora esistono numerosi siti web che permettono di consultare gli archivi dove sono contenute tali informazioni.

```

luigi@capogrosso ~ % whois univr.it
*****
* Please note that the following result could be a subgroup of      *
* the data contained in the database.                                *
*                                                               *
* Additional information can be visualized at:                      *
* http://web-whois.nic.it                                           *
* Privacy Information: http://web-whois.nic.it/privacy               *
*****



Domain:          univr.it
Status:          ok
Signed:          no
Created:         1996-01-29 00:00:00
Last Update:    2021-02-14 00:56:13
Expire Date:   2022-01-29

Registrant
Organization: Universita' di Verona
Address:        Via S.Francesco, 22
                Verona
                37129
                VR
                IT
Created:        2007-03-01 10:49:11
Last Update:   2011-03-24 11:01:07

Admin Contact
Name:           Giovanni Bianco
Address:        SIA - Servizi Informatici di Ateneo
                Via S.Francesco, 22
                Verona
                37129
                VR
                IT
Created:        2006-02-14 00:00:00
Last Update:   2011-03-24 11:01:08


*****



Technical Contacts
Name:          Alberto Manzoni
Address:        SIA - Servizi Informatici di Ateneo
                Via S.Francesco, 22
                Verona
                37129
                VR
                IT
Created:       2004-03-18 00:00:00
Last Update:  2011-03-24 11:01:09

Name:          Andrea Sartori
Address:        Via dell'Artigliere, 19
                Verona
                37129
                VR
                IT
Created:       2004-03-18 00:00:00
Last Update:  2019-03-06 10:55:13

Registrar
Organization: Consortium GARR
Name:          GARR-REG
Web:           http://www.garr.it
DNSSEC:        no

Nameservers
dns01.univr.it
dns02.univr.it
ns1.garr.net

```

9. Esercitazione

Per motivi di sicurezza non è possibile analizzare il traffico reale presente all'interno della rete del laboratorio e, quindi, per questa esercitazione, useremo del traffico precedentemente catturato.

Esercizio 1

Avviare Wireshark, aprire il menu **File/Open** e selezionare il file **capture.cap**.

Si prenda in considerazione il pacchetto numero 9 e si risponda alle seguenti domande/esercizi:

1. Che tipo di protocollo di livello Data-link è utilizzato? Come fa Wireshark a capirlo?
2. Disegnare la PDU di livello Data-link indicando il valore dei vari campi.
3. Qual è il MAC sorgente? Di che tipo è: unicast o broadcast?
4. Qual è il MAC destinazione? Di che tipo è: unicast o broadcast?
5. Che tipo di protocollo di livello Network è utilizzato? Come fa Wireshark a capirlo?
6. Qual è la lunghezza dell'header IP?
7. Quali sono gli indirizzi IP sorgente e destinazione?
8. Che tipo di protocollo di livello trasporto è contenuto in IP? Come fa Wireshark a capirlo?
9. Quali sono le porte sorgente e destinazione a livello trasporto?
10. Creare un filtro per visualizzare solo i pacchetti che hanno ARP come protocollo
(*suggerimento*: basta scrivere **arp** nella barra **Filter** sotto la toolbar; si ricordi di premere su **Apply** dopo aver scritto **arp**).
11. Dopo aver applicato il filtro precedente qual è la percentuale di pacchetti che rimangono visualizzati rispetto al totale?
(*suggerimento*: vedere entrambi i valori nella barra di stato in basso).
12. Creare un filtro per visualizzare solo i pacchetti che hanno destinazione MAC 00:01:e6:57:4b:e0.
(*suggerimento*: usare l'editor di espressioni; la categoria da selezionare è **Ethernet**; per

l'indirizzo MAC usare la notazione esadecimale con i due punti come separatori; si ricordi di premere su **Apply** dopo aver creato l'espressione).

13. Dopo aver applicato il filtro precedente qual è la percentuale di pacchetti che rimangono visualizzati rispetto al totale?
(*suggerimento:* vedere entrambi i valori nella barra di stato in basso).
14. Creare un filtro per visualizzare solo i pacchetti che hanno destinazione MAC broadcast.
(*suggerimento:* nell'editor di espressioni la categoria da usare è **Ethernet**; per l'indirizzo MAC usare la notazione esadecimale con i due punti come separatori; si ricordi di premere su **Apply** dopo aver creato l'espressione).
15. Dopo aver applicato il filtro precedente qual è la percentuale di pacchetti che rimangono visualizzati rispetto al totale? Sono molti? Perché?

Esercizio 2

Occorre aprire il menu **File/Open** e selezionare il file **simpleHTTP.cap**.

1. Colorare di rosso tutti i pacchetti che contengono UDP e di verde tutti i pacchetti che contengono TCP.
(*suggerimento:* nell'editor delle regole di colorazione è sufficiente portare in alto due regole già esistenti e modificarle per cambiarne i colori di sfondo).
2. Cosa contengono i primi due pacchetti della sessione di cattura?
 - IP sorgente, IP destinazione.
 - Tipo di protocollo di trasporto.
 - Tipo di protocollo di livello Applicazione. Come fa Wireshark a capirlo?
 - Messaggio contenuto nel Payload di livello applicazione.
3. Prendere in considerazione il pacchetto n. 3.
 - IP sorgente, IP destinazione.
 - Tipo di protocollo di trasporto.
 - IP sorgente e destinazione sono in qualche modo collegati con i messaggi scambiati a livello applicazione nei primi due pacchetti? È possibile fare delle ipotesi su cosa serve il protocollo di livello applicazione dei primi due pacchetti?
4. Prendere in considerazione il pacchetto n. 6.
 - IP sorgente, IP destinazione.
 - Tipo di protocollo di trasporto.
 - Tipo di protocollo di livello Applicazione.
 - Perché prima della trasmissione del primo messaggio HTTP c'è lo scambio di tre pacchetti puramente TCP? Quali sono i flag settati nell'header TCP di questi tre pacchetti?
4. Creare un filtro per visualizzare solo i pacchetti TCP (compresi i pacchetti HTTP) e determinarne il numero.
5. Creare un filtro per visualizzare solo i pacchetti TCP (esclusi i pacchetti HTTP) e determinarne il numero.
 - Qual è la percentuale sul totale dei pacchetti TCP trovata al punto 5?
 - A cosa servono tali pacchetti?
 - Se il protocollo DNS dei pacchetti 1 e 2 avesse usato il protocollo TCP, quanti pacchetti IP sarebbero stati generati? Sarebbe stato utile?
4. Selezionare il pacchetto 3 e seguire lo stream TCP col comando da menu **Analyze/Follow TCP Stream**.
 - Cosa si può leggere?
 - Qual è il messaggio contenuto nel payload della PDU di livello Applicazione?

Esercizio 3

Occorre aprire il menu **File/Open** e selezionare il file **busyNetwork.cap**.

1. Elencare i protocolli di livello Applicazione che entrano in azione in questa cattura classificandoli in base al livello Trasporto utilizzato.
2. Provare ad analizzare diversi stream TCP con sopra diversi protocolli di livello applicazione.
3. Che differenza c'è tra il contenuto trasmesso in una connessione TCP per il protocollo FTP e quello trasmesso per il protocollo SSH?

Esercizio 4

Occorre aprire il menu **File/Open** e selezionare il file **pingCapture.cap**.

1. Individuare le richieste ping inviate e le relative risposte. Quante sono?
2. Quali sono IP sorgente e destinazione della richiesta ICMP? A quale ente o azienda sono intestati?
3. Provare a invocare il comando **ping** dal proprio PC verso www.google.com e verso il proprio Default Gateway (come faccio a sapere il suo IP?) e osservare il RTT medio e la sua variazione. Chi mostra la media più grande? Perché? Chi mostra la variazione più grande? Perché?

Esercizio 5

Entrare nel sistema Linux presente in cloud e digitare il comando **traceroute www.google.com**

1. Individuare le interfacce dei router attraversati.
2. Individuare i nomi delle organizzazioni a cui sono intestati gli IP delle interfacce dei router attraversati.

Esercizio 6

1. Cercare quali interfacce sono attualmente attive sul proprio PC. Qual è l'indirizzo IP dell'interfaccia che state utilizzando sul vostro host? E la netmask corrispondente?
2. Qual è l'indirizzo IP di www.univr.it?