Lo strato di Rete
NAT
ICMP

Wireshark

Esercizio 2

Un host A ha l'indirizzo IP 167.199.170.82/27.

Indicare:

- l'indirizzo di rete di A
- il primo e ultimo indirizzo nel range degli indirizzi della rete di A

Esercizio 2 - soluzione

Un host A ha l'indirizzo IP 167.199.170.82/27.

Indicare:

- l'indirizzo di rete di A
- il primo e ultimo indirizzo nel range degli indirizzi della rete di A

[167] [199] [170] 01010010 01000000

167.199.170.64/27

Esercizio 2- soluzione

167.199.170.82/27

10100111 11000111 10101010 01010010

Indirizzi del blocco:

10100111 11000111 10101010 01000000

10100111 11000111 10101010 01000001

...

10100111 11000111 10101010 01011110

10100111 11000111 10101010 01011111

Esercizio 2 - soluzione

Address: 167.199.170.82/27

Mask: 255.255.255.224

Errore nel Forouzan a pag. 201

Primo indirizzo del blocco (indirizzo di rete) = address AND mask

```
  10100111  11000111  10101010  01010010
AND 11111111  11111111  11111111  11100000
  10100111  11000111  10101010  01000000
```

167.99.170.64

Ultimo indirizzo del blocco (broadcast) = address OR NOT(mask)

```
  10100111  11000111  10101010  01010010
OR  00000000  00000000  00000000  00011111
  10100111  11000111  10101010  01011111
```

167.199.170.95

Network Address Translation

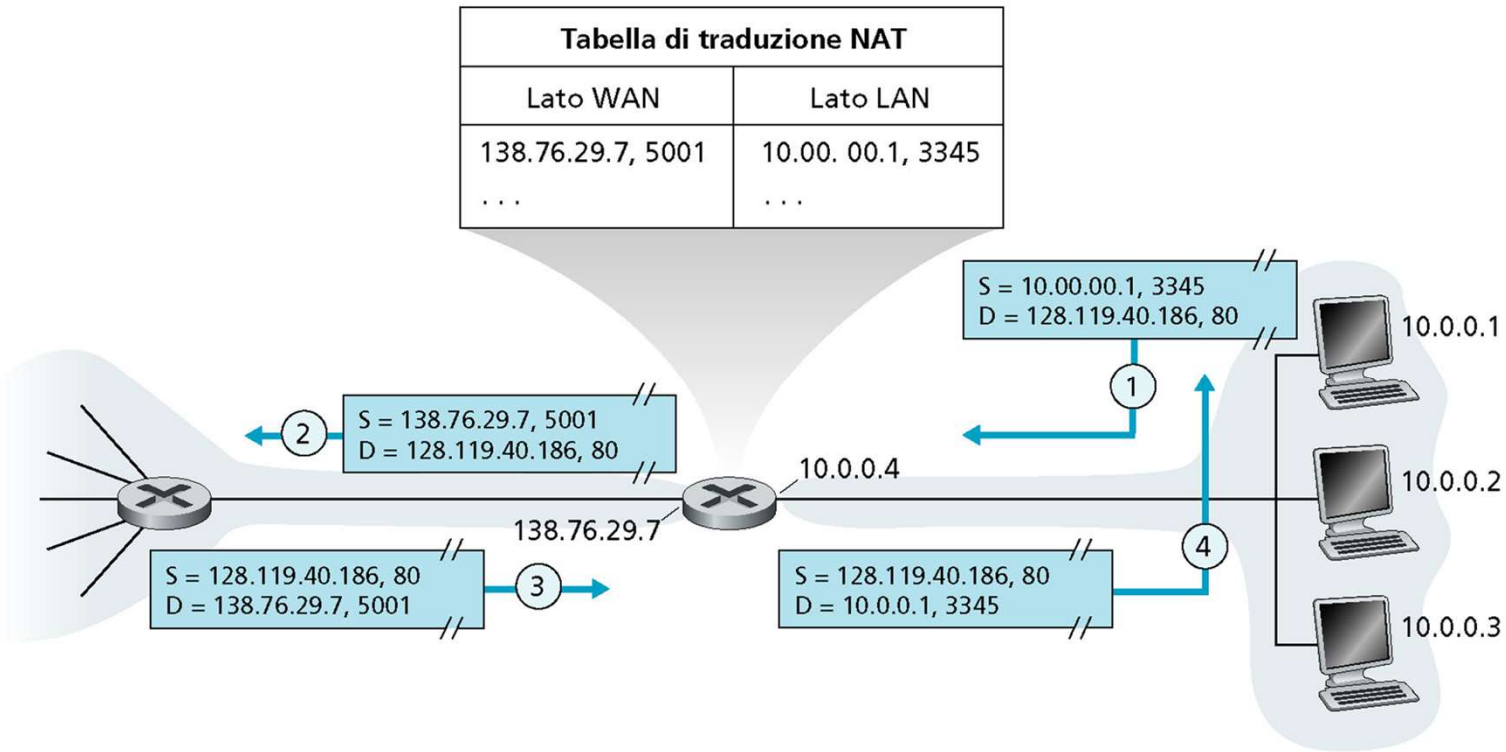
Come abilitare il traffico verso/dalla rete Internet pubblica per una rete privata?

- NAT ovvero Network Address Translation (RFC 2663,3022), permette di trasmettere su Internet il traffico proveniente da sistemi attestati su sottoreti private, in cui sono assegnati indirizzi IP privati
- L'accesso di una rete privata su Internet è realizzato attraverso un router abilitato alla NAT, questo router ha almeno un indirizzo IP pubblico
  - Tutto il traffico (datagram) in uscita dal router di accesso ha un indirizzo IP sorgente pubblico, quello del router.
  - Tutto il traffico in ingresso alla subnet ha come indirizzo IP di destinazione l'indirizzo pubblico del router di accesso.

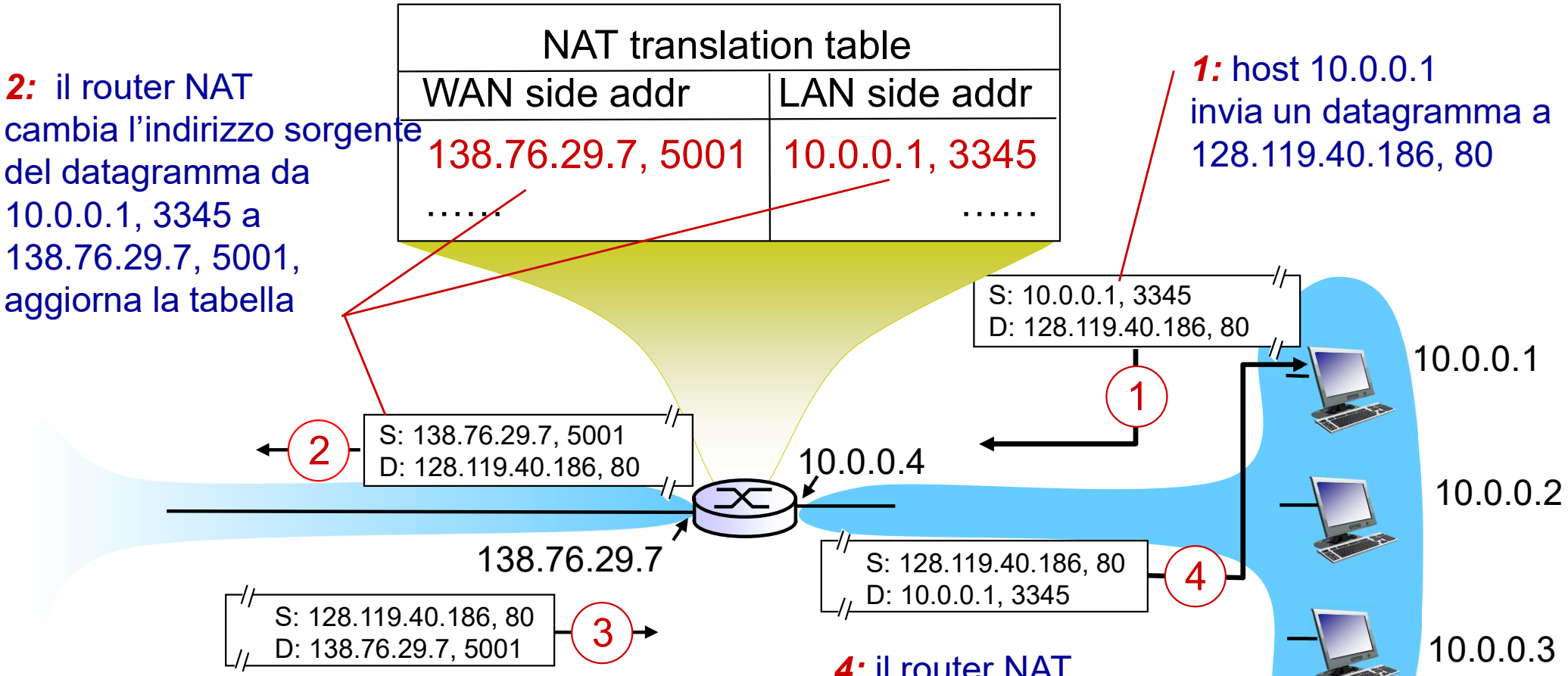
Network Address Translation (II)

Il router di accesso ha in memoria una tabella di traduzione NAT, le cui righe contengono le associazioni

`(IP privato, porta)(IP pubblico e porta assegnata dal router)`



NAT: network address translation



Riflessioni su NAT

- Quante comunicazioni può gestire un router NAT con un solo indirizzo pubblico?

Approfondimenti

- Cosa succede se un client FTP sotto NAT invia RETR (e router NAT del client FTP non accetta richieste di connessioni TCP)?
- In una applicazione P2P come può un peer chiedere di utilizzare una certa porta per poter essere contattato?
- E se volessi rendere disponibile un server web ospitato su un host della sottorete privata?

ICMP

- Vi sono situazioni nelle quali si è verificato un errore (es. nodo non raggiungibile), ma il protocollo IP non ha meccanismi integrati per renderlo noto all'host mittente.
- L'IPv4 non implementa alcun meccanismo per segnalare gli errori o correggerli.
- Cosa accade se qualcosa va storto?
- Inoltre, il protocollo IP è sprovvisto di un meccanismo per effettuare richieste sullo stato di un sistema remoto.

- L'Internet Control Message Protocol versione 4 (ICMPv4) è stato creato per porre rimedio a queste carenze.

ICMP

- Usato da host e router per scambiarsi messaggi di errore o altre situazioni che richiedono intervento.
- Messaggi ICMP sono incapsulati all'interno di datagrammi IP.
  - Ma viene comunque considerato parte integrante dello strato di rete
  - chi implementa IP deve supportare anche ICMP.
- RFC 792

ICMP

- Quando un router o un host di destinazione devono informare il mittente di errori o di eventi avvenuti nell'inoltro di un pacchetto IP utilizzano il protocollo Internet Control Message Protocol (ICMP).
  - Es. Messaggio di errore: Port is unreachable
- I pacchetti ICMP vengono instradati dai router prima dei pacchetti IP ordinari
  - Per pacchetti IP frammentati, i messaggi ICMP sono relativi al solo frammento con offset 0 (il primo frammento).
  - I messaggi ICMP NON sono mai inviati in risposta a pacchetti IP con indirizzo mittente che non rappresenta in modo univoco un host (es. 0.0.0.0, 127.0.0.1).
  - I messaggi ICMP NON sono mai inviati in risposta a pacchetti IP con destinazione IP broadcast o multicast (o link broadcast)
- I messaggi ICMP non sono mai inviati in risposta a messaggi di errore ICMP, ma possono essere inviati in risposta a messaggi ICMP di interrogazione.

Tipi di messaggio ICMP

- I messaggi si distinguono in:
  - Messaggi di segnalazione errore
  - Messaggi di richiesta/risposta
- Nell'header ICMP i campi Tipo (8 bit) e Codice (8 bit) indicano tipo e significato dei messaggi

Tipi di messaggio ICMP

ICMP Tipo	Codice	Descrizione
0	0	risposta al messaggio di eco (a ping) - *echo replay*
3	0	rete di destinazione irraggiungibile - *destination network unreachable*
3	1	host di destinazione irraggiungibile - *destination host unreachable*
3	2	protocollo di destinazione irraggiungibile - *destination protocol unreachable*
3	3	porta di destinazione irraggiungibile - *destination port unreachable*
3	6	rete di destinazione sconosciuta - *destination network unknown*
3	7	host di destinazione sconosciuto - *destination host unreachable*
4	0	strozzamento della sorgente (controllo della congestione) - *source quench*
8	0	richiesta di eco - *echo request*
9	0	annuncio dal router - *router advertisement*
10	0	scoperta del router - *router discovery*
11	0	TTL scaduto - *TTL expired*
12	0	cattiva intestazione IP - *IP header bad*

Ping

- Una delle applicazioni che un host può utilizzare per verificare il funzionamento di un altro host è il programma ping.

- Il programma ping si basa sui messaggi di richiesta e risposta eco dell'ICMP.
  - Un host invia una richiesta eco (tipo 8, codice 0) a un altro host che, se attivo, può rispondere con una risposta eco (tipo 0, codice 0).
- Fornisce anche misure dell'RTT

- In maniera molto grossolana, il programma ping può anche misurare l'affidabilità e la congestione del router tra due host inviando una sequenza di messaggi richiesta-risposta.

Ping

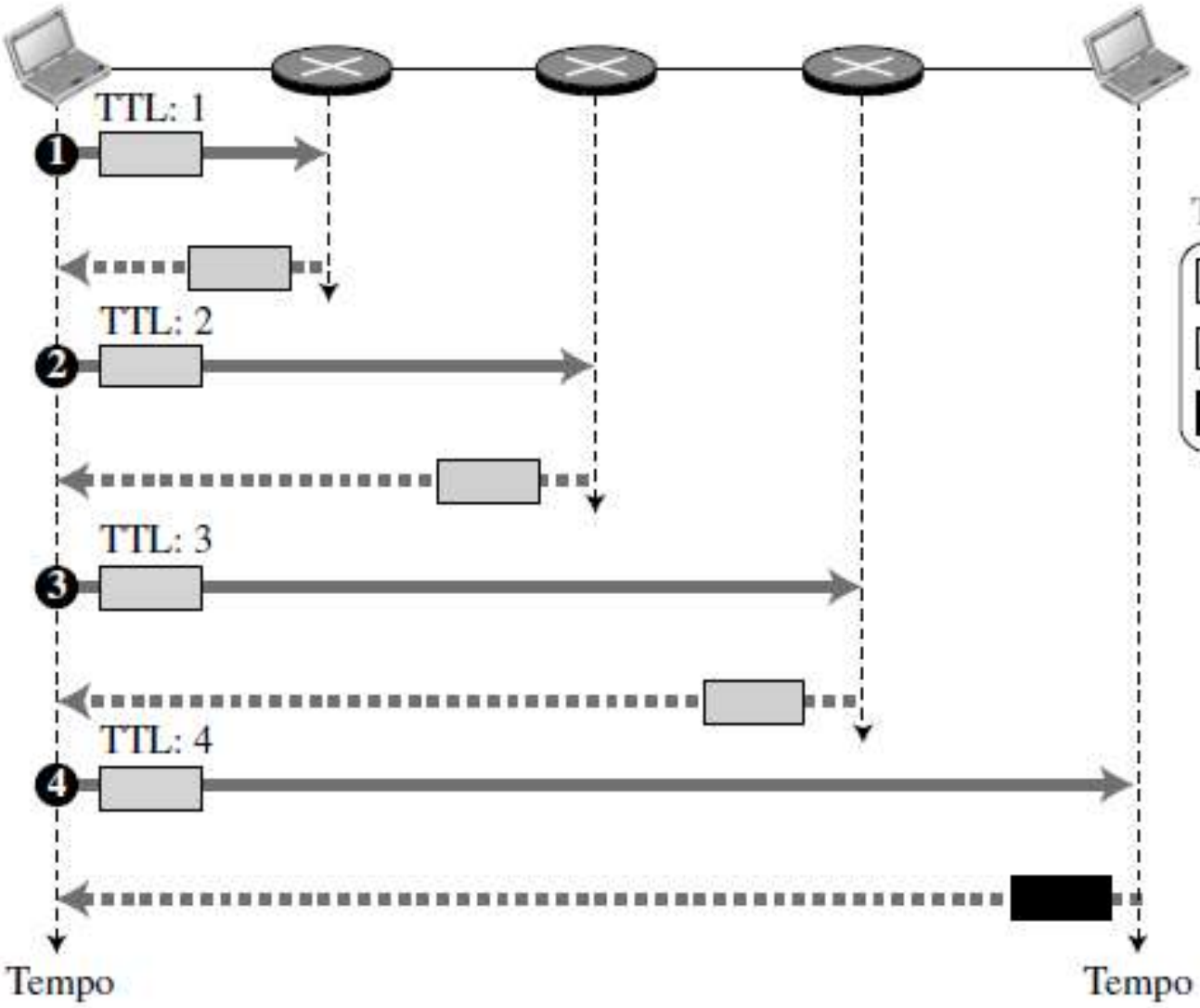
Invio di un messaggio ping al sito pads.cs.unibo.it:

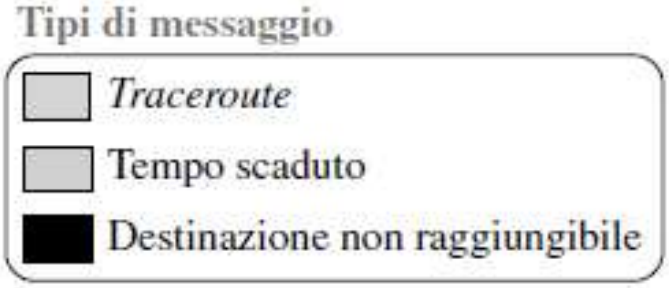
```
$ ping pads.cs.unibo.it
PING chernobog.pads.cs.unibo.it (130.136.132.11) 56(84) bytes of data.
64 bytes from chernobog.pads.cs.unibo.it (130.136.132.11): icmp_req=1 ttl=52 time=34.2 ms
64 bytes from chernobog.pads.cs.unibo.it (130.136.132.11): icmp_req=2 ttl=52 time=33.1 ms
64 bytes from chernobog.pads.cs.unibo.it (130.136.132.11): icmp_req=3 ttl=52 time=34.0 ms
64 bytes from chernobog.pads.cs.unibo.it (130.136.132.11): icmp_req=4 ttl=52 time=33.9 ms
64 bytes from chernobog.pads.cs.unibo.it (130.136.132.11): icmp_req=5 ttl=52 time=33.3 ms
--- chernobog.pads.cs.unibo.it ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 33.177/33.758/34.220/0.417 ms
```

Traceroute

- Il programma traceroute in UNIX o tracert in Windows può essere utilizzato per individuare il percorso di un datagramma dalla sorgente alla destinazione tramite l'identificazione dell'indirizzo IP di tutti i router che vengono visitati lungo il percorso.
- Solitamente il programma viene impostato per un massimo di 30 salti (router), che sono usualmente sufficienti per raggiungere la destinazione.
- Implementazione di traceroute
  - Uso del protocollo UDP, si invia un datagramma ad un porta su cui è improbabile che un processo sia in ascolto;
  - i datagrammi sono configurati in modo da scatenare l'invio di messaggi di errore di due tipi:
    - tempo scaduto
    - Porta destinazione non raggiungibile

Traceroute





Invio di n +1 datagrammi UDP (n numero di router nel percorso) con valori di TTL crescenti

Quando TTL=0 il router invia un messaggio di errore di *tempo scaduto*

Il messaggio di errore inviato dall'host destinazione è *port unreachable*

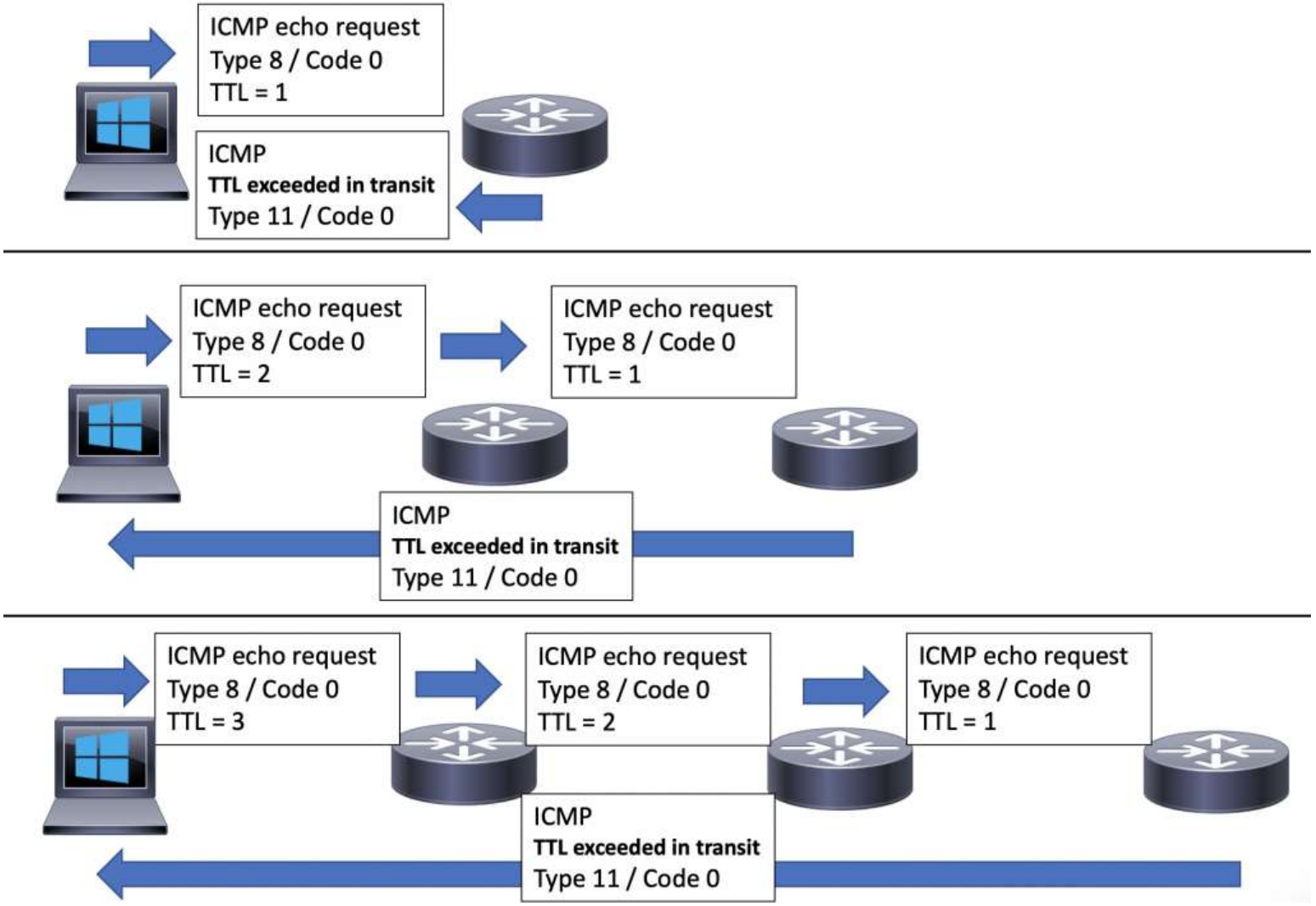
Traceroute

- Il programma traceroute imposta inoltre un timer per trovare il tempo di round-trip di ciascun router e della destinazione.
- La maggior parte dei programmi traceroute invia tre messaggi a ogni dispositivo, con lo stesso valore di TTL, per poter effettuare una stima migliore del tempo di round-trip.
- Quanto segue mostra un esempio di funzionamento del programma traceroute che utilizza tre messaggi per ogni dispositivo e ottiene quindi tre RTT:

```
$ traceroute printers.com
traceroute to printers.com (13.1.69.93), 30 hops max, 38 byte packets
1 route.front.edu (153.18.31.254) 0.622 ms 0.891 ms 0.875 ms
2 ceneric.net (137.164.32.140) 3.069 ms 2.875 ms 2.930 ms
3 satire.net (132.16.132.20) 3.071 ms 2.876 ms 2.929 ms
4 alpha.printers.com (13.1.69.93) 5.922 ms 5.048 ms 4.922 ms
```

Traceroute

- L'implementazione del comando tracert su windows usa messaggi ICMP echo request/reply



ICMP & Wireshark

1. Avviare wireshark e iniziate la cattura dei pacchetti
2. Su un terminale digitare

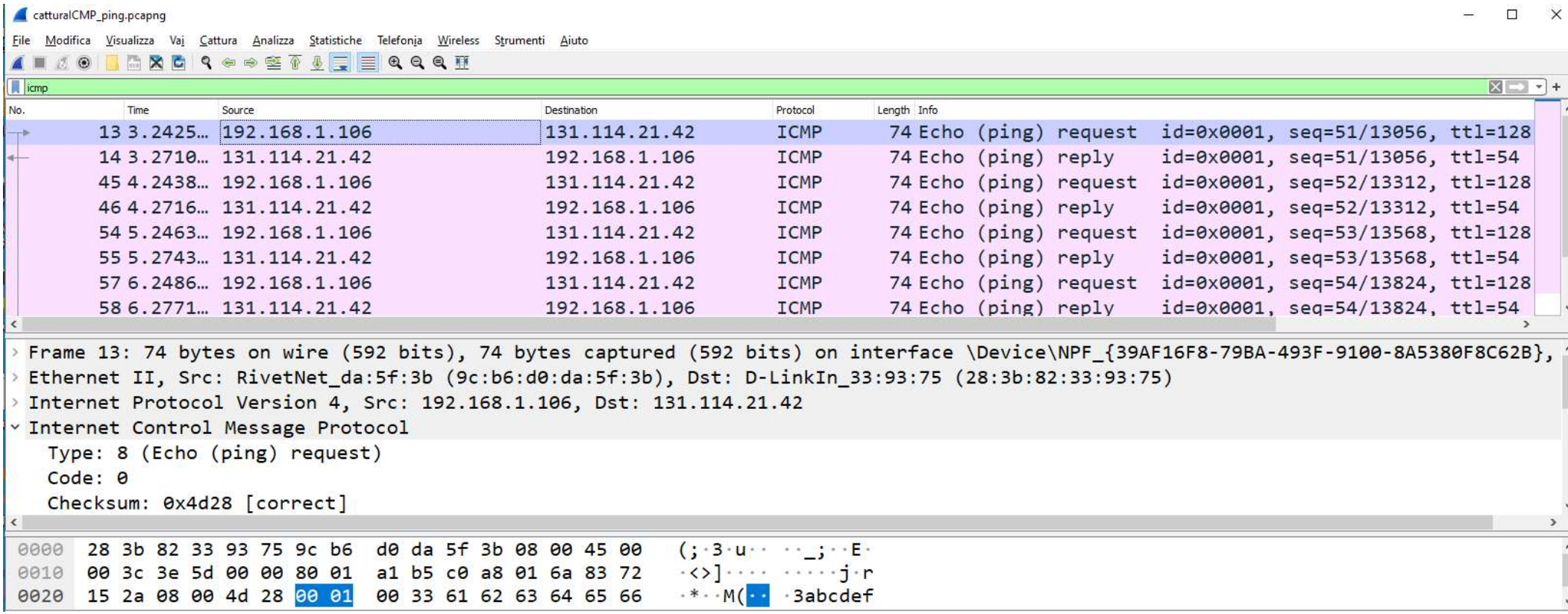
   `ping –n 5 www.unipi.it`

   -n opzione per indicare il numero di pacchetti da inviare
3. Quando il programma ping termina, interrompere la cattura dei pacchetti

```
[federica.Dell] ▸ ping -n 5 www.unipi.it

Esecuzione di Ping wwwnew2.unipi.it [131.114.21.42] con 32 byte di dati:
Risposta da 131.114.21.42: byte=32 durata=28ms TTL=54
Risposta da 131.114.21.42: byte=32 durata=30ms TTL=54
Risposta da 131.114.21.42: byte=32 durata=28ms TTL=54
Risposta da 131.114.21.42: byte=32 durata=30ms TTL=54
Risposta da 131.114.21.42: byte=32 durata=29ms TTL=54

Statistiche Ping per 131.114.21.42:
    Pacchetti: Trasmessi = 5, Ricevuti = 5,
    Persi = 0 (0% persi),
Tempo approssimativo percorsi andata/ritorno in millisecondi:
    Minimo = 28ms, Massimo =  30ms, Medio =  29ms
```

ICMP & Wireshark

- Su wireshark inserite icmp nel campo filtro di visualizzazione



Traceroute & Wireshark

1. Avviare wireshark e iniziate la cattura dei pacchetti
2. Su un terminale digitare

   `tracert` www.corriere.it (NB windows usa echo request, non UDP datagrams)

   Su linux `traceroute www.corriere.it`
3. Quando il programma traceroute termina, interrompere la cattura dei pacchetti

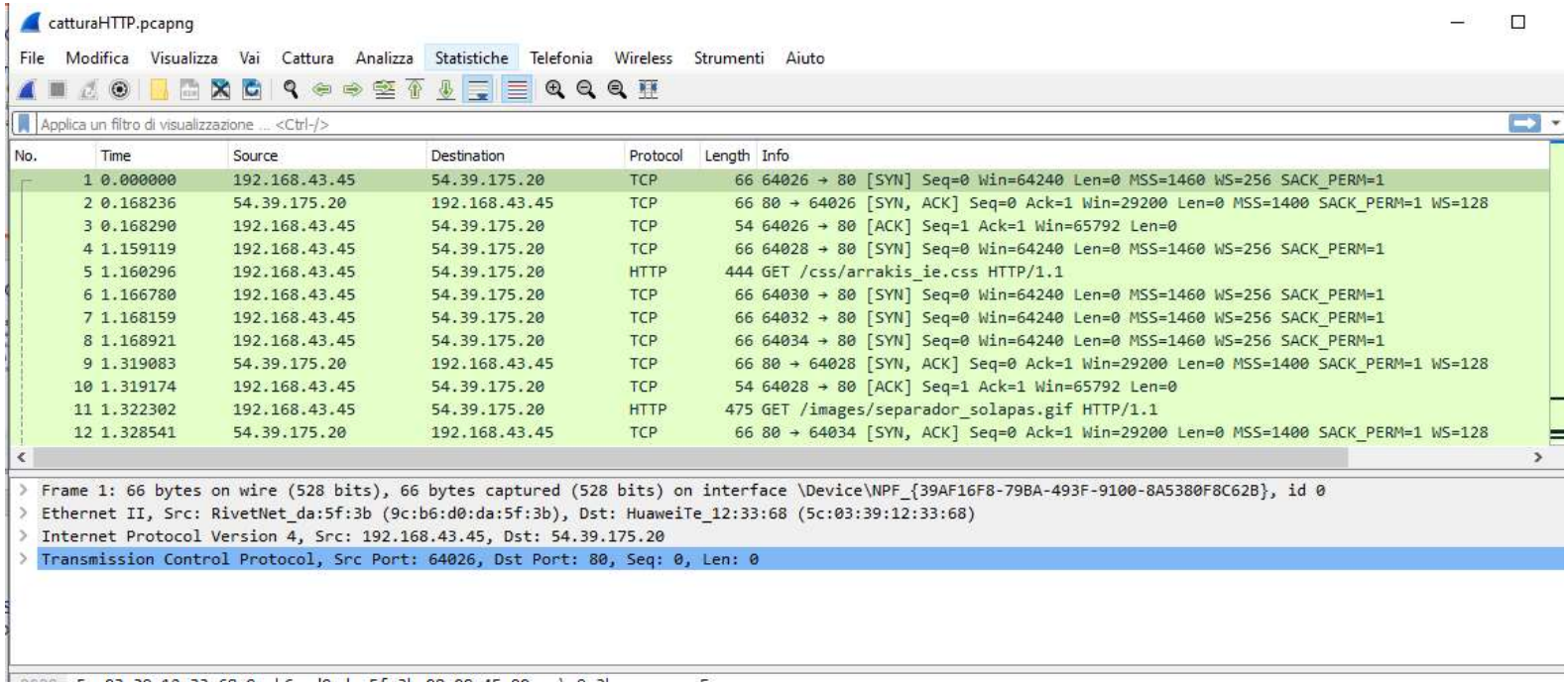
```
[federica.Dell] ➤ tracert www.corriere.it

Traccia instradamento verso na-eu-corriere.map.fastly.net [151.101.113.50]
su un massimo di 30 punti di passaggio:

  1     2 ms     3 ms     2 ms  192.168.1.1
  2    11 ms    11 ms    15 ms  151.7.198.6
  3    12 ms    12 ms    11 ms  151.7.32.136
  4    11 ms    41 ms    12 ms  151.7.32.160
  5    21 ms    17 ms    16 ms  151.6.0.91
  6    18 ms    18 ms    16 ms  151.6.0.159
  7     *        *        *     Richiesta scaduta.
  8    25 ms    25 ms    26 ms  151.101.113.50

Traccia completata.
```

Wireshark

Cos’è Wireshark?

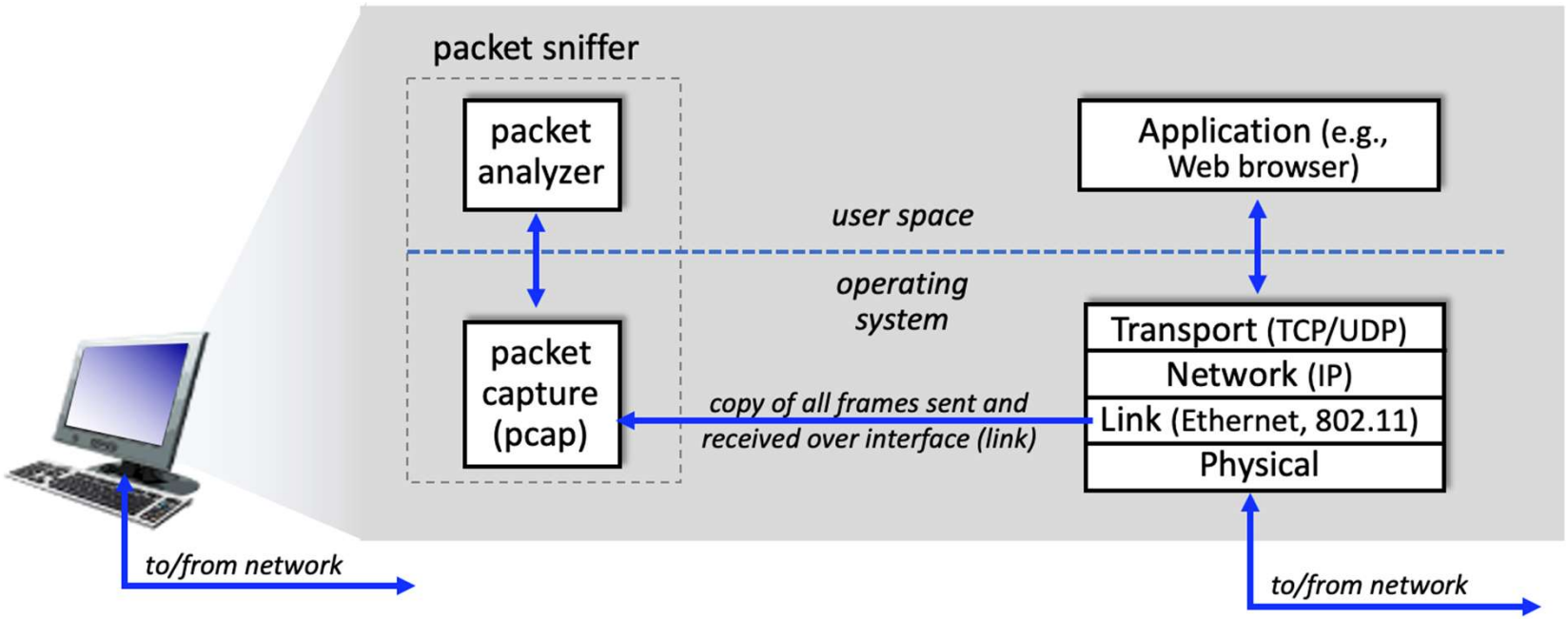
- Analizzatore di protocolli di rete
- Wireshark è un software libero
- Per Windows, Linux/Unix e Mac OS

https://www.wireshark.org/



Come funziona?

- Libreria di cattura dei pacchetti: copia passivamente (ossia "sniffa, annusa") i messaggi che vengono inviati e ricevuti da/a una interfaccia di rete del vostro computer (usa le librerie *libpcap* o *WinPCap*)
- Analizzatore di pacchetti: visualizza il contenuto di tutti i camp



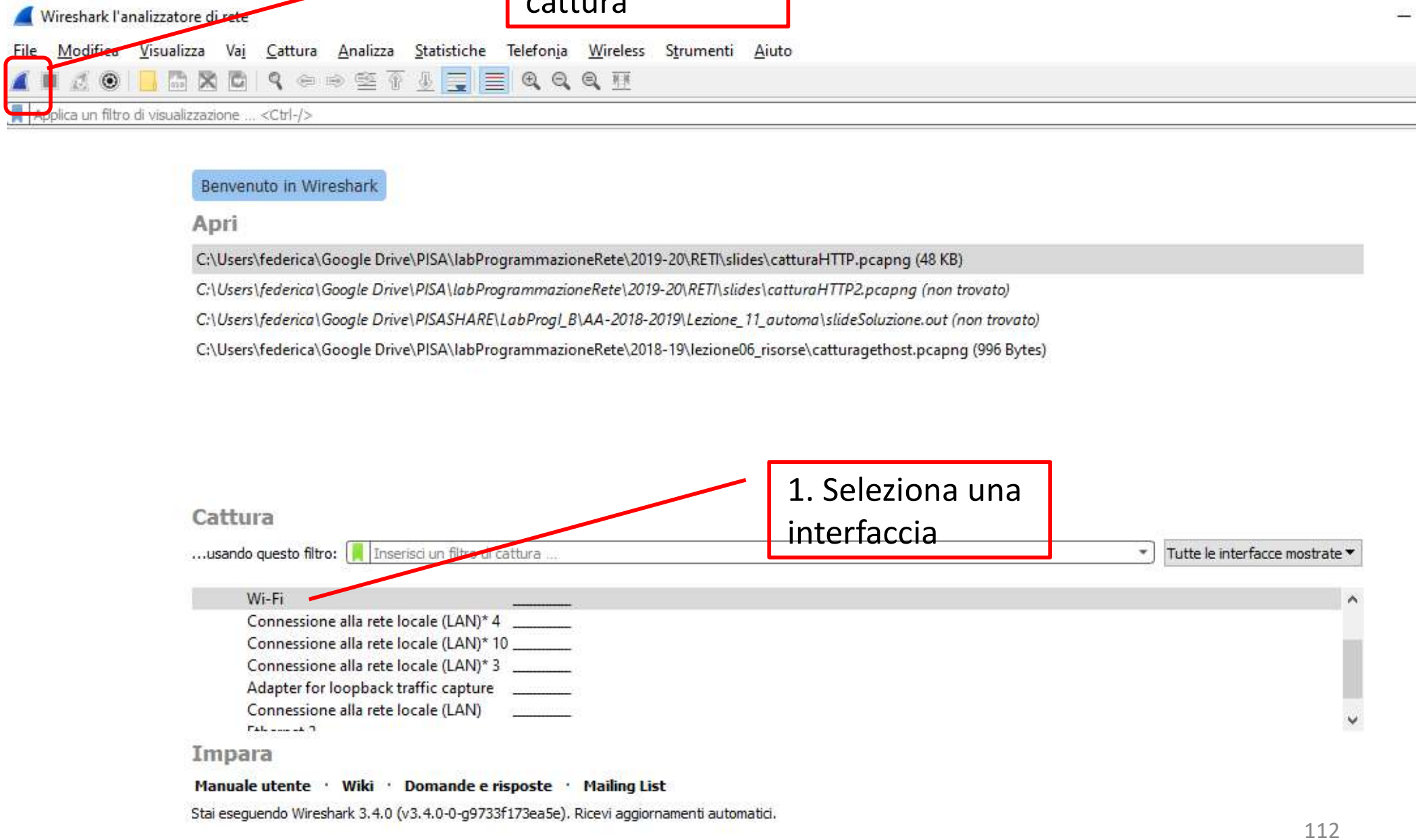
Per scaricarlo

- Per quanto riguarda il sistema operativo Linux, Wireshark è incluso in pressoché qualunque distribuzione, per cui è sufficiente installarlo come si fa per gli altri pacchetti software

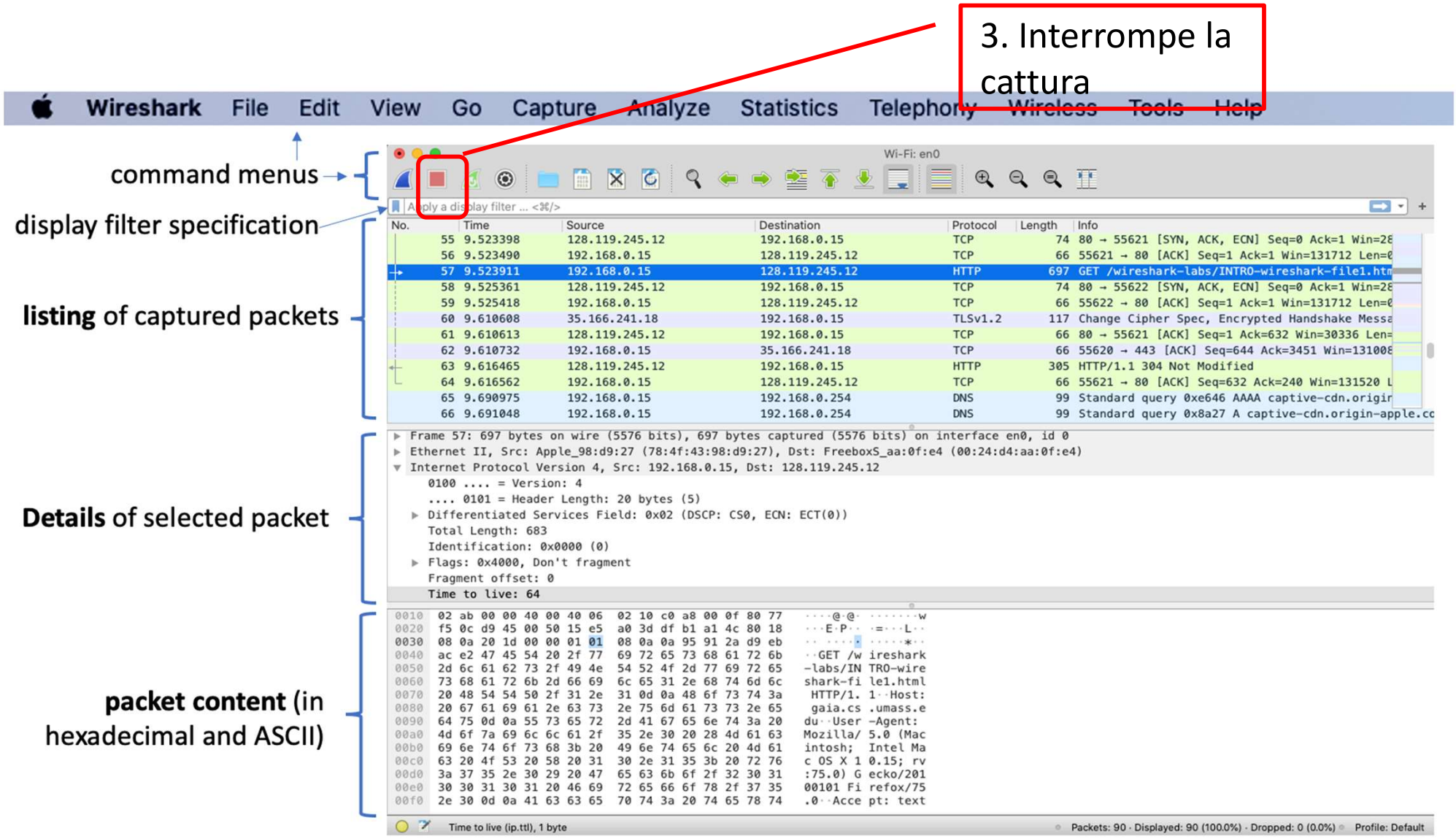
```
sudo apt-get update
sudo apt-get install wireshark
sudo wireshark (per avviarlo)
```

- Per gli altri sistemi operativi, è possibile scaricare il codice eseguibile di Wireshark dalla pagina http://www.wireshark.org/download.html

Interfaccia



Interfaccia



Interfaccia

- Il **menù dei comandi** è un menù a discesa collocato in cima alla finestra. Il menù File vi consente di salvare i dati catturati, aprire un file contenente dati precedentemente catturati e uscire da Wireshark. Il menù Capture consente di iniziare la cattura dei pacchetti.
- La **finestra di elenco dei pacchetti** mostra un riassunto di una linea per ogni pacchetto catturato, incluso il numero di pacchetto (che è un numero assegnato da Wireshark; *non* è contenuto nella intestazione di alcun protocollo), il tempo al quale il pacchetto è stato catturato, gli indirizzi sorgente e destinazione, il tipo di protocollo, e informazioni specifiche del protocollo contenuto nel pacchetto.
- La **finestra di dettaglio delle intestazioni** mostra dettagli sul pacchetto selezionato nell'elenco dei pacchetti catturati.
- La **finestra di contenuto del pacchetto** mostra l'intero contenuto del frame catturato, sia in forma esadecimale che ASCII.
- **filtro sui pacchetti da visualizzare**, all'interno del quale è possibile digitare un nome di protocollo o altre informazioni per *filtrare* i pacchetti da visualizzare nell'elenco dei pacchetti catturati (e quindi anche nelle finestre del dettaglio e dei contenuti). Nell'esempio

Esempio con HTTP

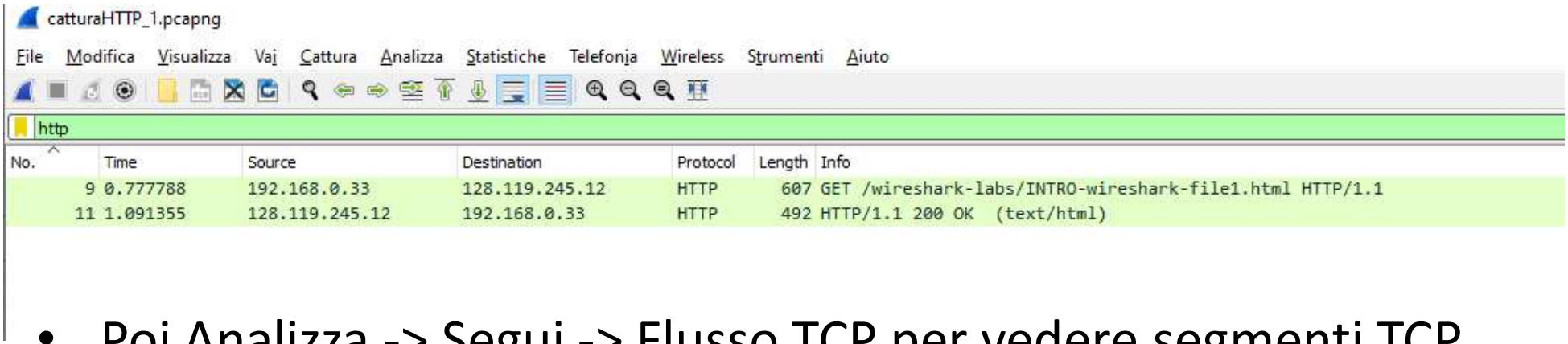
- Mentre Wireshark è in esecuzione e sta catturando i pacchetti
- immettete nel browser la URL

http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html

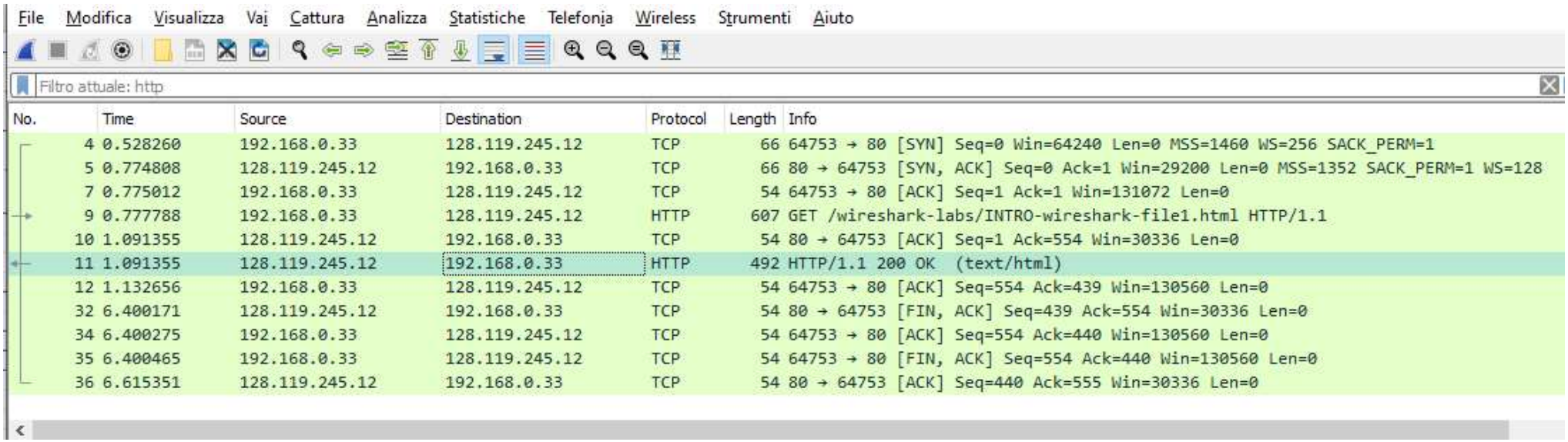
- Dopo che il browser ha visualizzato la pagina INTRO-wire1.html, interrompete la cattura dei pacchetti selezionando Stop nella finestra riassuntiva di cattura
- Opzione: Cattura -> Filtri -> http

Esempio con HTTP

- Digita *http* sul filtro di cattura



- Poi Analizza -> Segui -> Flusso TCP per vedere segmenti TCP



Esempio con FTP

- Ripetere i passi dell'esempio con anonymouys ftp
- Sul filtro scrivere *ftp*
  - Pacchetti relativi alla connessione di controllo
- Per vedere tutto il flusso

  ip.addr== 129.215.17.244 (o comunque IP server ftp
- Per selezionare la connessione dati, seleziona segmento con FTP-DATA, poi vai su Analizza->Segui-> Segui Flusso TCP

Riferimenti

- Laboratorio Wireshark: Introduzione Versione 6.0 italiano © 2005-2012 J.F. Kurose, K. W. Ross. All rights reserved
- Wireshark Lab: Getting Started v8.0 Supplement to *Computer Networking: A Top-Down Approach, 8th ed.,* J.F. Kurose and K.W. Ross