

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

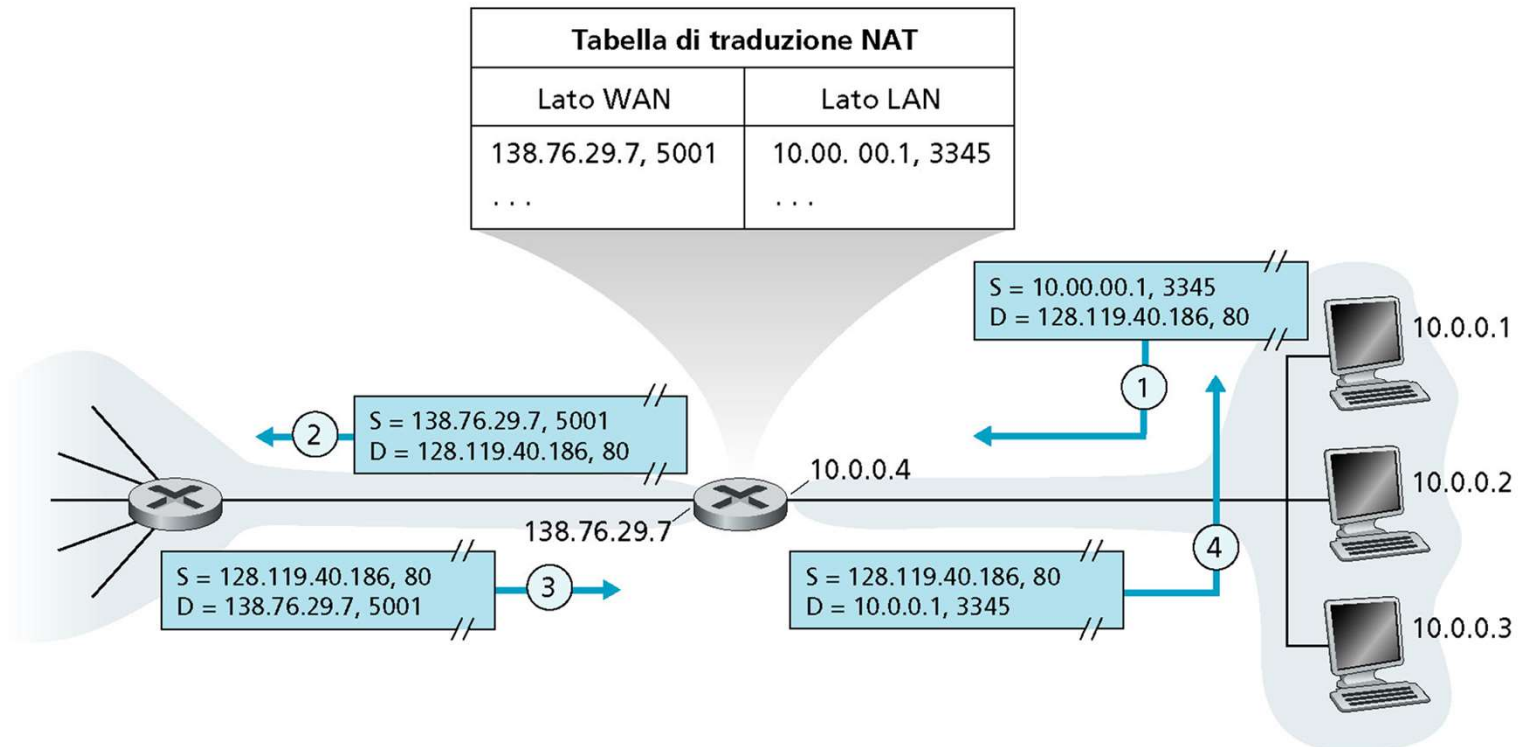
Come abilitare il traffico verso/dalla rete Internet pubblica per una rete privata?

- 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.

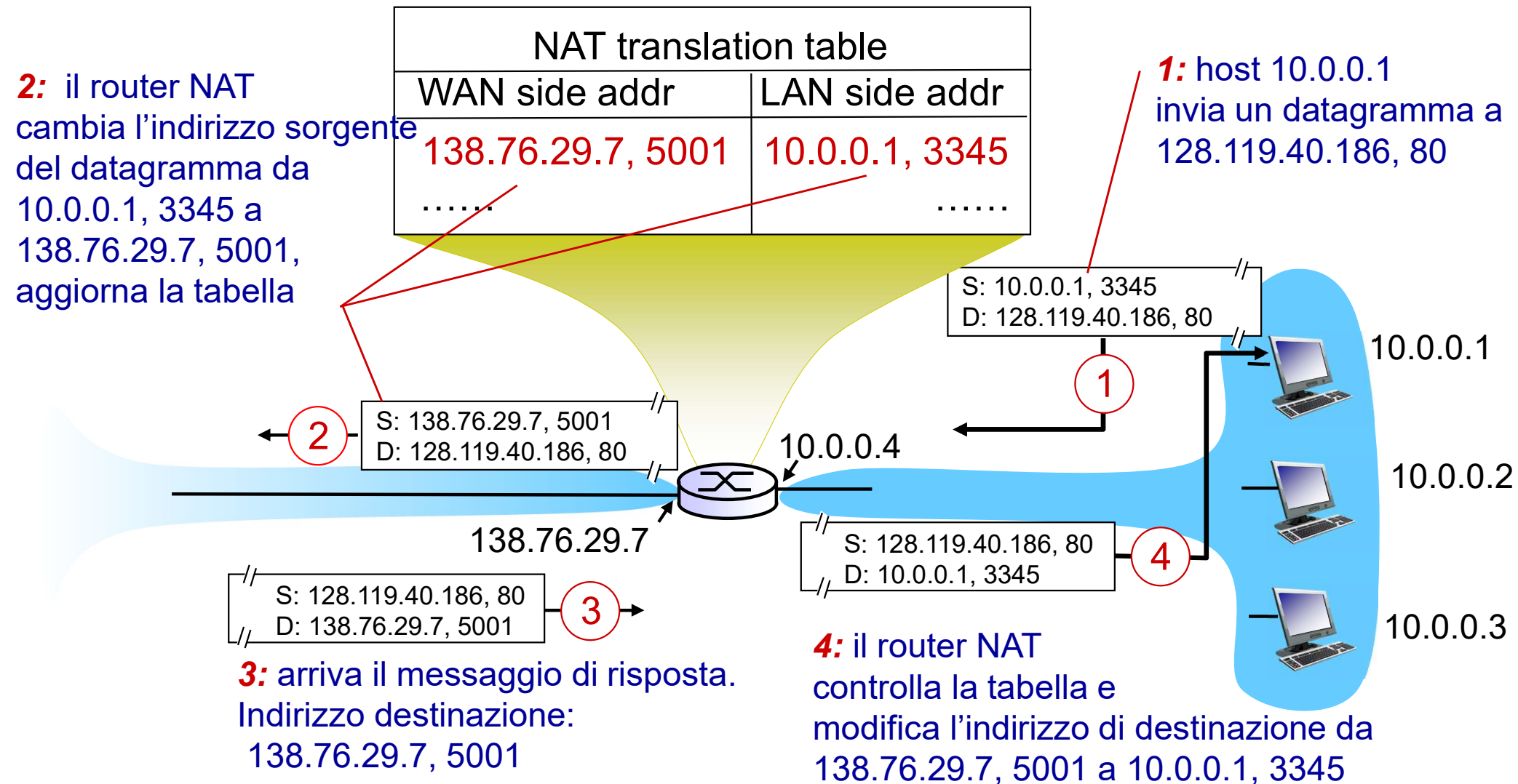
Network Address Translation (II)

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

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

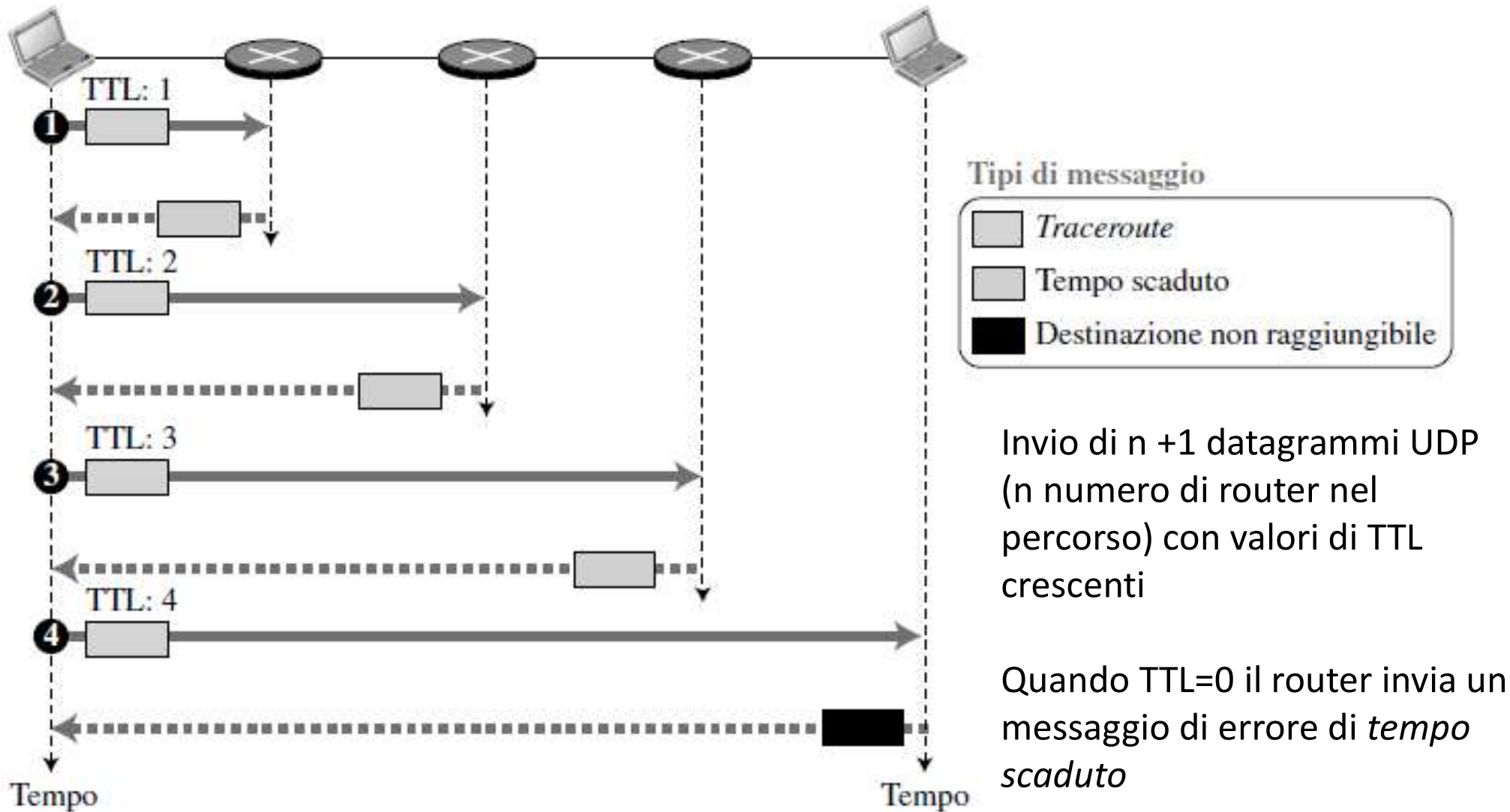
Invio di un messaggio ping al sito pads.cs.unibo.it:

```
$ 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 una 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



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:

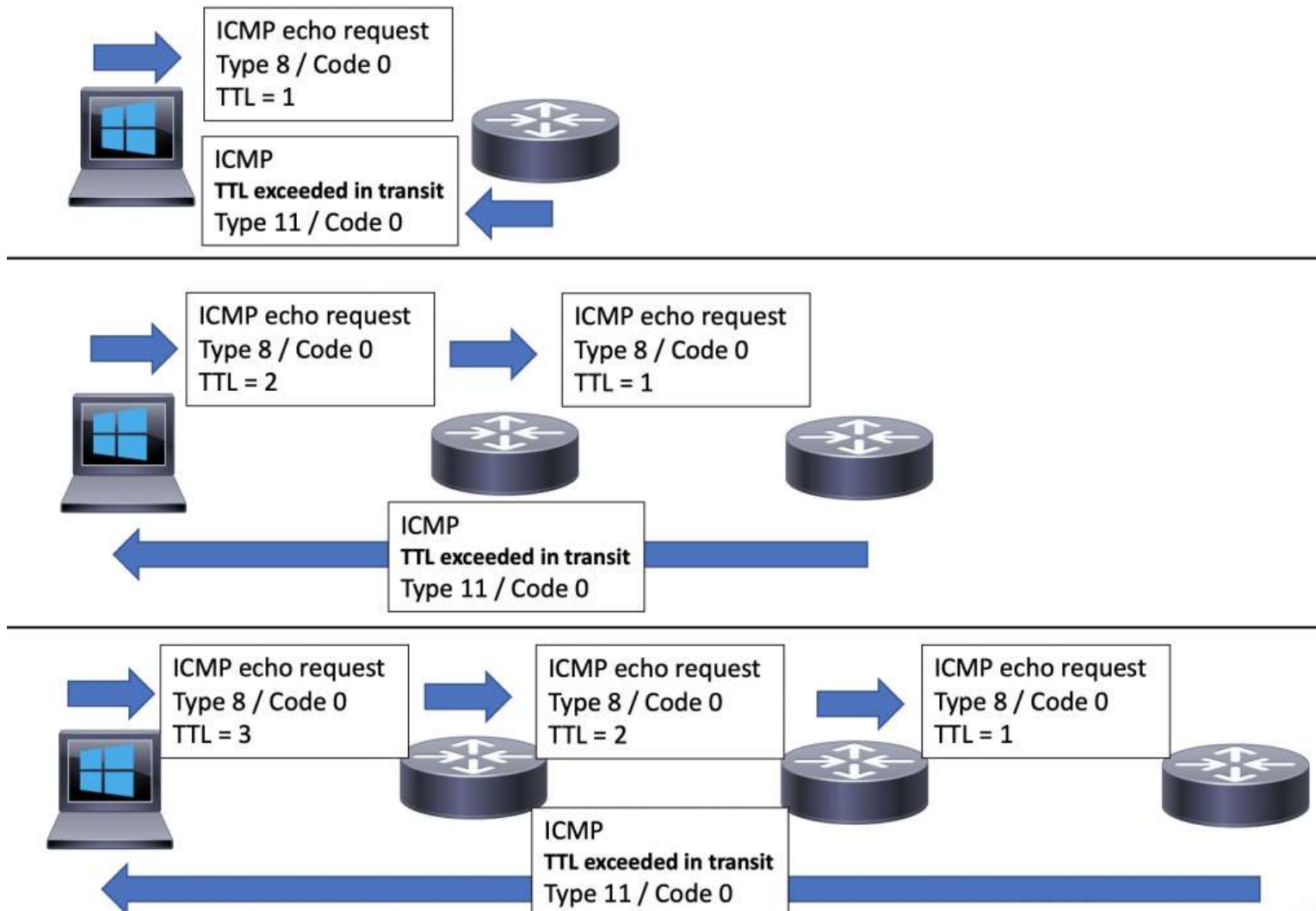
```
$ traceroute printers.com
```

```
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 iniziare 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

The image shows a Wireshark capture of ICMP ping traffic. The packet list pane displays eight packets, alternating between requests and replies. The packet details pane for packet 13 shows the ICMP Echo (ping) request structure, including the type, code, and checksum. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
13	3.2425...	192.168.1.106	131.114.21.42	ICMP	74	Echo (ping) request id=0x0001, seq=51/13056, ttl=128
14	3.2710...	131.114.21.42	192.168.1.106	ICMP	74	Echo (ping) reply id=0x0001, seq=51/13056, ttl=54
45	4.2438...	192.168.1.106	131.114.21.42	ICMP	74	Echo (ping) request id=0x0001, seq=52/13312, ttl=128
46	4.2716...	131.114.21.42	192.168.1.106	ICMP	74	Echo (ping) reply id=0x0001, seq=52/13312, ttl=54
54	5.2463...	192.168.1.106	131.114.21.42	ICMP	74	Echo (ping) request id=0x0001, seq=53/13568, ttl=128
55	5.2743...	131.114.21.42	192.168.1.106	ICMP	74	Echo (ping) reply id=0x0001, seq=53/13568, ttl=54
57	6.2486...	192.168.1.106	131.114.21.42	ICMP	74	Echo (ping) request id=0x0001, seq=54/13824, ttl=128
58	6.2771...	131.114.21.42	192.168.1.106	ICMP	74	Echo (ping) reply id=0x0001, seq=54/13824, ttl=54

Frame 13: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{39AF16F8-79BA-493F-9100-8A5380F8C62B},
> Ethernet II, Src: RivetNet_da:5f:3b (9c:b6:d0:da:5f:3b), Dst: D-LinkIn_33:93:75 (28:3b:82:33:93:75)
> Internet Protocol Version 4, Src: 192.168.1.106, Dst: 131.114.21.42
Internet Control Message Protocol
Type: 8 (Echo (ping) request)
Code: 0
Checksum: 0x4d28 [correct]

```
0000  28 3b 82 33 93 75 9c b6 d0 da 5f 3b 08 00 45 00  ( ; 3 u . . . _ ; . E .
0010  00 3c 3e 5d 00 00 80 01 a1 b5 c0 a8 01 6a 83 72  . < > ] . . . . . . . . j . r
0020  15 2a 08 00 4d 28 00 01 00 33 61 62 63 64 65 66  . * . . M ( . . . 3 abcdef
```

Traceroute & Wireshark

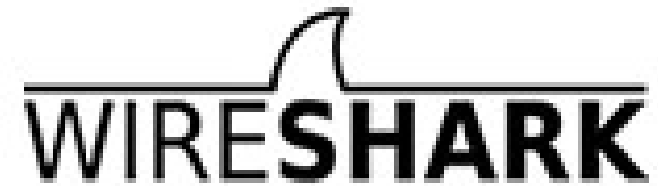
1. Avviare wireshark e iniziare la cattura dei pacchetti
2. Su un terminale digitare
tracert www.corriere.it (NB windows usa echo request, non UDP datagrams)
Su linux traceroute www.corriere.it
3. Quando il programma traceroute termina, interrompere la cattura dei pacchetti

```
[2020-11-15 22:28:25]  
[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.
```

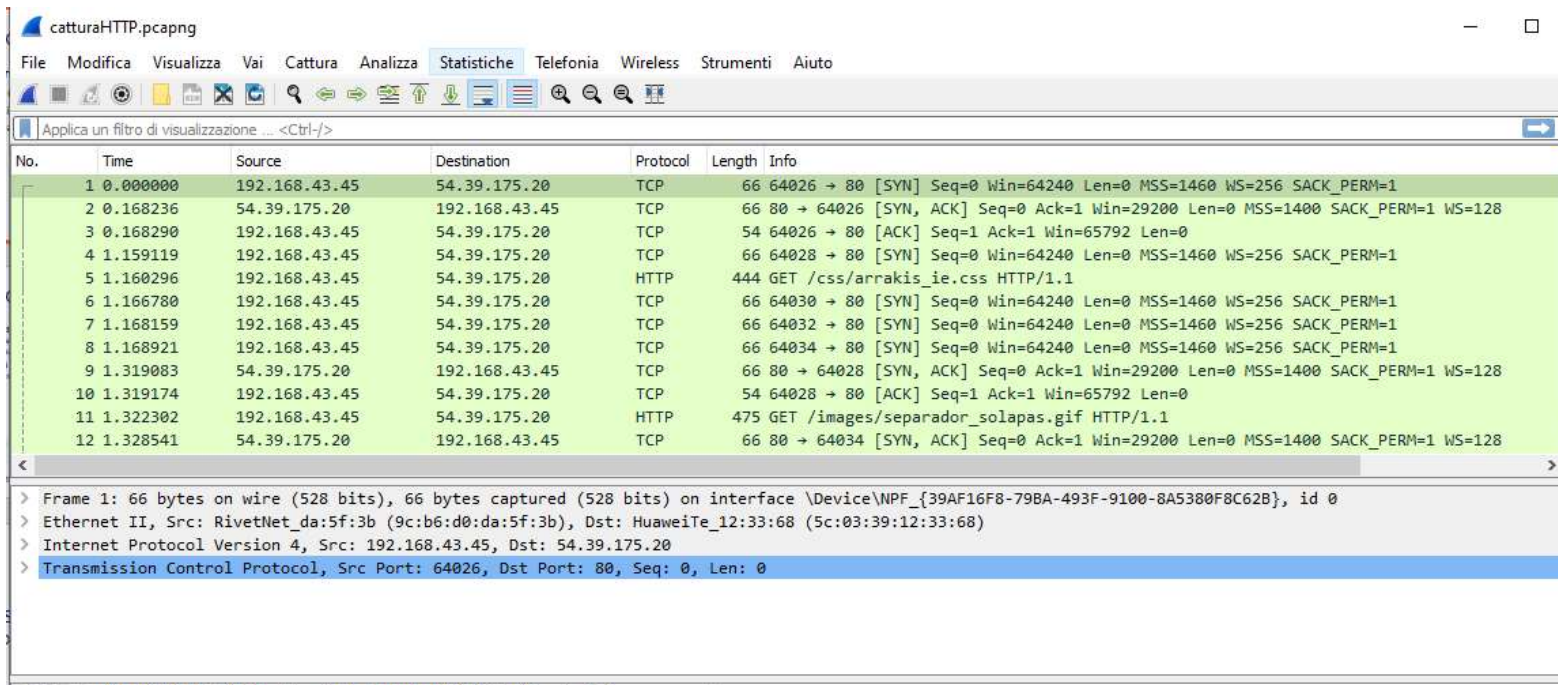
Wireshark

Cos'è Wireshark?

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

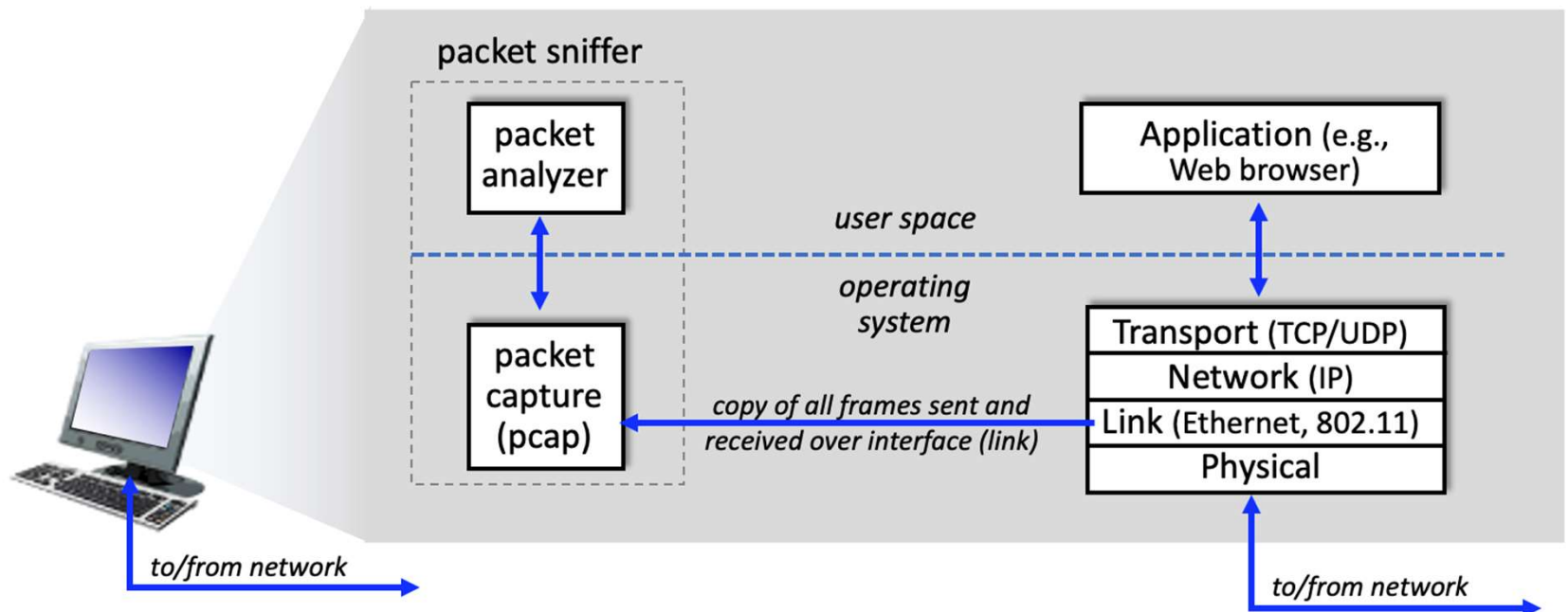


<https://www.wireshark.org/>



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

2. Fai partire la
cattura



Benvenuto in Wireshark

Apri

C:\Users\federica\Google Drive\PISA\labProgrammazioneRete\2019-20\RETI\slides\catturaHTTP.pcapng (48 KB)
C:\Users\federica\Google Drive\PISA\labProgrammazioneRete\2019-20\RETI\slides\catturaHTTP2.pcapng (non trovato)
C:\Users\federica\Google Drive\PISASHARE\LabProgl_B\AA-2018-2019\Lezione_11_automa\slideSoluzione.out (non trovato)
C:\Users\federica\Google Drive\PISA\labProgrammazioneRete\2018-19\lezione06_risorse\catturagethost.pcapng (996 Bytes)

1. Seleziona una
interfaccia

Cattura



Impara

[Manuale utente](#) · [Wiki](#) · [Domande e risposte](#) · [Mailing List](#)

Stai eseguendo Wireshark 3.4.0 (v3.4.0-0-g9733f173ea5e). Ricevi aggiornamenti automatici.

Interfaccia

3. Interrompe la
cattura

The image displays the Wireshark network protocol analyzer interface. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. The toolbar contains various icons for file operations, capture control, and analysis. A red box highlights the 'Stop Capture' button (a red square) in the toolbar, with a red arrow pointing to the text '3. Interrompe la cattura'.

On the left side, blue arrows point to different parts of the interface with labels:

- command menus**: points to the menu bar.
- display filter specification**: points to the 'Apply a display filter' text box.
- listing of captured packets**: points to the packet list table.
- Details of selected packet**: points to the packet details pane.
- packet content (in hexadecimal and ASCII)**: points to the packet bytes pane.

The main window shows a list of captured packets. The selected packet (No. 57) is an HTTP GET request. The details pane shows the structure of the packet, including Ethernet II, Internet Protocol Version 4, and Hypertext Transfer Protocol. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
55	9.523398	128.119.245.12	192.168.0.15	TCP	74	80 → 55621 [SYN, ACK, ECN] Seq=0 Ack=1 Win=28
56	9.523490	192.168.0.15	128.119.245.12	TCP	66	55621 → 80 [ACK] Seq=1 Ack=1 Win=131712 Len=0
57	9.523911	192.168.0.15	128.119.245.12	HTTP	697	GET /wireshark-labs/INTRO-wireshark-file1.htm
58	9.525361	128.119.245.12	192.168.0.15	TCP	74	80 → 55622 [SYN, ACK, ECN] Seq=0 Ack=1 Win=28
59	9.525418	192.168.0.15	128.119.245.12	TCP	66	55622 → 80 [ACK] Seq=1 Ack=1 Win=131712 Len=0
60	9.610608	35.166.241.18	192.168.0.15	TLSv1.2	117	Change Cipher Spec, Encrypted Handshake Messa
61	9.610613	128.119.245.12	192.168.0.15	TCP	66	80 → 55621 [ACK] Seq=1 Ack=632 Win=30336 Len=0
62	9.610732	192.168.0.15	35.166.241.18	TCP	66	55620 → 443 [ACK] Seq=644 Ack=3451 Win=131008
63	9.616465	128.119.245.12	192.168.0.15	HTTP	305	HTTP/1.1 304 Not Modified
64	9.616562	192.168.0.15	128.119.245.12	TCP	66	55621 → 80 [ACK] Seq=632 Ack=240 Win=131520 L
65	9.690975	192.168.0.15	192.168.0.254	DNS	99	Standard query 0xe646 AAAA captive-cdn.origin
66	9.691048	192.168.0.15	192.168.0.254	DNS	99	Standard query 0x8a27 A captive-cdn.origin-apple.cc

Details of selected packet (Frame 57):

- Frame 57: 697 bytes on wire (5576 bits), 697 bytes captured (5576 bits) on interface en0, id 0
- Ethernet II, Src: Apple_98:d9:27 (78:4f:43:98:d9:27), Dst: FreeboxS_aa:0f:e4 (00:24:d4:aa:0f:e4)
- Internet Protocol Version 4, Src: 192.168.0.15, Dst: 128.119.245.12
 - 0100 = Version: 4
 - 0101 = Header Length: 20 bytes (5)
 - Differentiated Services Field: 0x02 (DSCP: CS0, ECN: ECT(0))
 - Total Length: 683
 - Identification: 0x0000 (0)
 - Flags: 0x4000, Don't fragment
 - Fragment offset: 0
 - Time to live: 64

Packet content (hexadecimal and ASCII):

```
0010 02 ab 00 00 40 00 40 06 02 10 c0 a8 00 0f 80 77  ....@.  ....w
0020 f5 0c d9 45 00 50 15 e5 a0 3d df b1 a1 4c 80 18  ...E.P.  ...=...L..
0030 08 0a 20 1d 00 00 01 01 08 0a 0a 95 91 2a d9 eb  ....*...
0040 ac e2 47 45 54 20 2f 77 69 72 65 73 68 61 72 6b  .GET /w ireshark
0050 2d 6c 61 62 73 2f 49 4e 54 52 4f 2d 77 69 72 65  -labs/IN TRO-wire
0060 73 68 61 72 6b 2d 66 69 6c 65 31 2e 68 74 6d 6c  shark-fi le1.html
0070 20 48 54 54 50 2f 31 2e 31 0d 0a 48 6f 73 74 3a  HTTP/1.1 -Host:
0080 20 67 61 69 61 2e 63 73 2e 75 6d 61 73 73 2e 65  gaia.cs .umass.e
0090 64 75 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20  du..User -Agent:
00a0 4d 6f 7a 69 6c 6c 61 2f 35 2e 30 20 28 4d 61 63  Mozilla/ 5.0 (Mac
00b0 69 6e 74 6f 73 68 3b 20 49 6e 74 65 6c 20 4d 61 intosh; Intel Ma
00c0 63 20 4f 53 20 58 20 31 30 2e 31 35 3b 20 72 76  c OS X 1 0.15; rv
00d0 3a 37 35 2e 30 29 20 47 65 63 6b 6f 2f 32 30 31  :75.0) Gecko/201
00e0 30 30 31 30 31 20 46 69 72 65 66 6f 78 2f 37 35 00101 F1 refox/75
00f0 2e 30 0d 0a 41 63 63 65 70 74 3a 20 74 65 78 74  .0..Acce pt: text
```


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

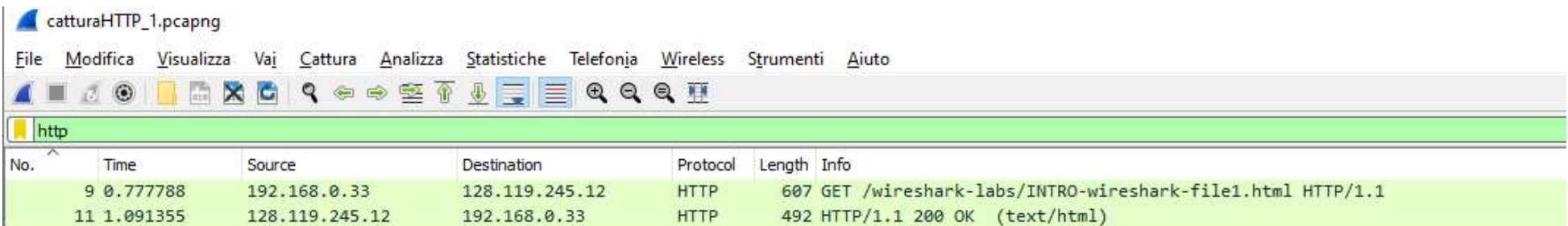
- Mentre Wireshark è in esecuzione e sta catturando i pacchetti
- immettete nel browser la URL

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

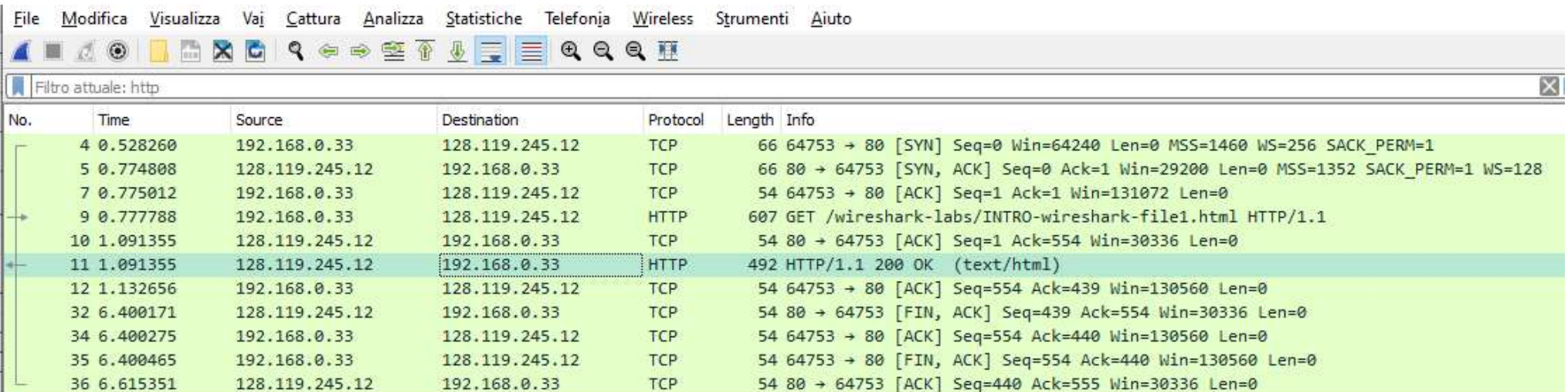
- Dopo che il browser ha visualizzato la pagina INTRO-wire1.html, interrompete la cattura dei pacchetti selezionando Stop nella finestra riassuntiva di cattura
- Opzione: Cattura -> Filtri -> http

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 `anonymous 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