Packet Analysis

Programmazione di Reti

Ing. Chiara Contoli, PhD
chiara.contoli@unibo.it

Corso di Laurea Triennale in
Ingegneria e Scienze Informatiche

Protocol Analyzer

- Tools for packet analysis of data arriving at the network interface card:
  - Tcpdump
  - Wireshark
- Packet header view
- Allow to
  - Debug network applications
  - Debug network itself

Tcpdump

- CLI (Command Line Interface) only
- Multiple parameters can be specified to define what you want to monitor
- Most common:
  - *-n*: do not resolve DNS name
  - *-v* (o *-vv*): verbose output
  - *-c* <n>: capture n packets
  - *-i* <Intf>: capture packets on Intf interface
- *Filters*: show only traffic that satisfy certain characteristics
  # tcpdump -q -n -c 10 ‘host 192.168.1.10 and port ssh’

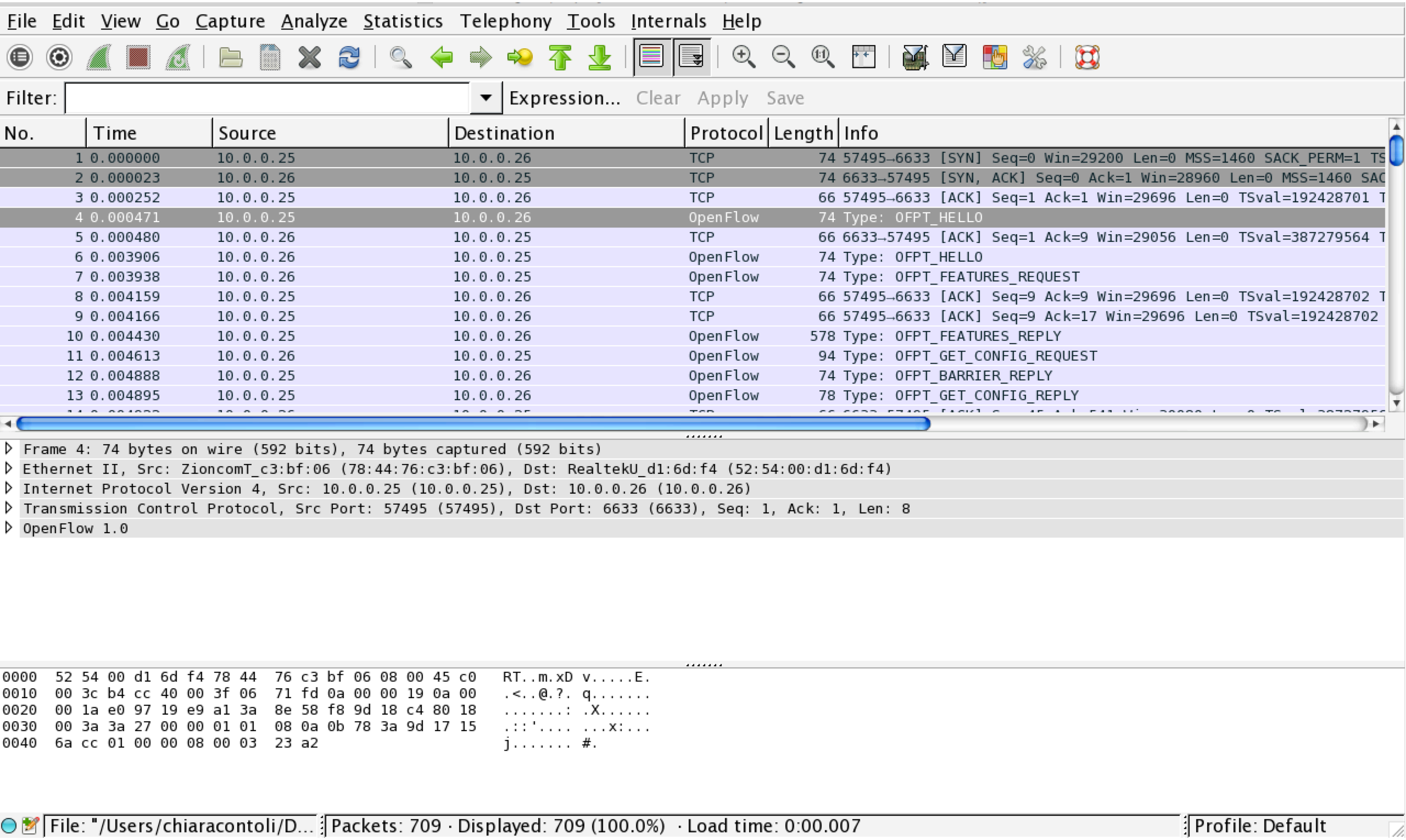
Tcpdump: example

```
mininet@mininet-cluster3:~$ sudo tcpdump -nve –i eth3 icmp
tcpdump: listening on eth3, link-type EN10MB (Ethernet), capture size 65535 bytes
08:17:17.051282 00:00:00:00:00:01 > 00:00:00:00:00:03, ethertype IPv4 (0x0800), length 98: (tos 0x0,
ttl 64, id 13417, offset 0, flags [DF], proto ICMP (1), length 84)
    192.168.1.1 > 172.16.2.100: ICMP echo request, id 13352, seq 1, length 64
08:17:17.079350 00:00:00:00:00:03 > 00:00:00:00:00:01, ethertype IPv4 (0x0800), length 98: (tos 0x0,
ttl 59, id 43350, offset 0, flags [none], proto ICMP (1), length 84)
    172.16.2.100 > 192.168.1.1: ICMP echo reply, id 13352, seq 1, length 64
08:17:18.052525 00:00:00:00:00:01 > 00:00:00:00:00:03, ethertype IPv4 (0x0800), length 98: (tos 0x0,
ttl 64, id 13418, offset 0, flags [DF], proto ICMP (1), length 84)
    192.168.1.1 > 172.16.2.100: ICMP echo request, id 13352, seq 2, length 64
08:17:18.059675 00:00:00:00:00:03 > 00:00:00:00:00:01, ethertype IPv4 (0x0800), length 98: (tos 0x0,
ttl 59, id 43565, offset 0, flags [none], proto ICMP (1), length 84)
    172.16.2.100 > 192.168.1.1: ICMP echo reply, id 13352, seq 2, length 64
08:17:19.053860 00:00:00:00:00:01 > 00:00:00:00:00:03, ethertype IPv4 (0x0800), length 98: (tos 0x0,
ttl 64, id 13419, offset 0, flags [DF], proto ICMP (1), length 84)
    192.168.1.1 > 172.16.2.100: ICMP echo request, id 13352, seq 3, length 64
08:17:19.059410 00:00:00:00:00:03 > 00:00:00:00:00:01, ethertype IPv4 (0x0800), length 98: (tos 0x0,
ttl 59, id 43638, offset 0, flags [none], proto ICMP (1), length 84)
    172.16.2.100 > 192.168.1.1: ICMP echo reply, id 13352, seq 3, length 64
…..
…..
```

Wireshark

- GUI (Graphical User Interface) tool
- Allow to specify filters based on several criteria
- Most common criteria
  - Protocols
  - Addresses
  - Ports
  - Specific fields (e.g., flags)

Wireshark: example



```
No.   Time      Source       Destination  Protocol Length Info
    1 0.000000  10.0.0.25    10.0.0.26    TCP        74 57495→6633 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TS
    2 0.000023  10.0.0.26    10.0.0.25    TCP        74 6633→57495 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SAC
    3 0.000252  10.0.0.25    10.0.0.26    TCP        66 57495→6633 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=192428701 T
    4 0.000471  10.0.0.25    10.0.0.26    OpenFlow   74 Type: OFPT_HELLO
    5 0.000480  10.0.0.26    10.0.0.25    TCP        66 6633→57495 [ACK] Seq=1 Ack=9 Win=29056 Len=0 TSval=387279564 T
    6 0.003906  10.0.0.26    10.0.0.25    OpenFlow   74 Type: OFPT_HELLO
    7 0.003938  10.0.0.26    10.0.0.25    OpenFlow   74 Type: OFPT_FEATURES_REQUEST
    8 0.004159  10.0.0.25    10.0.0.26    TCP        66 57495→6633 [ACK] Seq=9 Ack=9 Win=29696 Len=0 TSval=192428702 T
    9 0.004166  10.0.0.25    10.0.0.26    TCP        66 57495→6633 [ACK] Seq=9 Ack=17 Win=29696 Len=0 TSval=192428702
   10 0.004430  10.0.0.25    10.0.0.26    OpenFlow  578 Type: OFPT_FEATURES_REPLY
   11 0.004613  10.0.0.26    10.0.0.25    OpenFlow   94 Type: OFPT_GET_CONFIG_REQUEST
   12 0.004888  10.0.0.25    10.0.0.26    OpenFlow   74 Type: OFPT_BARRIER_REPLY
   13 0.004895  10.0.0.25    10.0.0.26    OpenFlow   78 Type: OFPT_GET_CONFIG_REPLY
```

```
▷ Frame 4: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
▷ Ethernet II, Src: ZioncomT_c3:bf:06 (78:44:76:c3:bf:06), Dst: RealtekU_d1:6d:f4 (52:54:00:d1:6d:f4)
▷ Internet Protocol Version 4, Src: 10.0.0.25 (10.0.0.25), Dst: 10.0.0.26 (10.0.0.26)
▷ Transmission Control Protocol, Src Port: 57495 (57495), Dst Port: 6633 (6633), Seq: 1, Ack: 1, Len: 8
▷ OpenFlow 1.0
```

```
0000  52 54 00 d1 6d f4 78 44  76 c3 bf 06 08 00 45 c0   RT..m.xD v.....E.
0010  00 3c b4 cc 40 00 3f 06  71 fd 0a 00 00 19 0a 00   .<..@.?. q.......
0020  00 1a e0 97 19 e9 a1 3a  8e 58 f8 9d 18 c4 80 18   .......: .X......
0030  00 3a 3a 27 00 00 01 01  08 0a 0b 78 3a 9d 17 15   .::'.... ...x:...
0040  6a cc 01 00 00 08 00 03  23 a2                     j....... #.
```

File: "/Users/chiaracontoli/D... Packets: 709 · Displayed: 709 (100.0%) · Load time: 0:00.007 Profile: Default

Other useful tools

- TCP/IP parameters configuration
- Connectivity check

Command (Linux / Windows)	Description
ifconfig / ipconfig	Check network configuration
ping <host>	Connectivity check towards *host*
traceroute -n <host> / *tracert -d* <host>	(Not only) Connectivity check towards *host*
route -n / route print	Routing table status

FTP Analysis
(File Transfer Protocol)

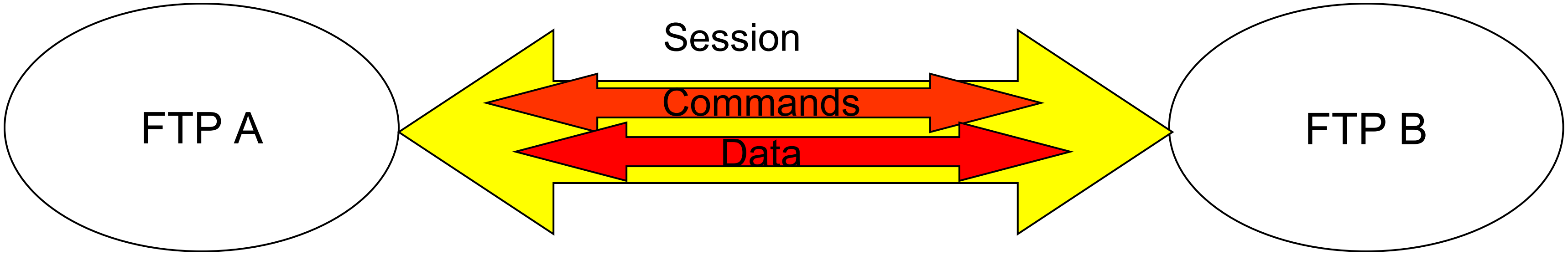
Programmazione di Reti

Ing. Chiara Contoli, PhD
chiara.contoli@unibo.it

Corso di Laurea Triennale in
Ingegneria e Scienze Informatiche

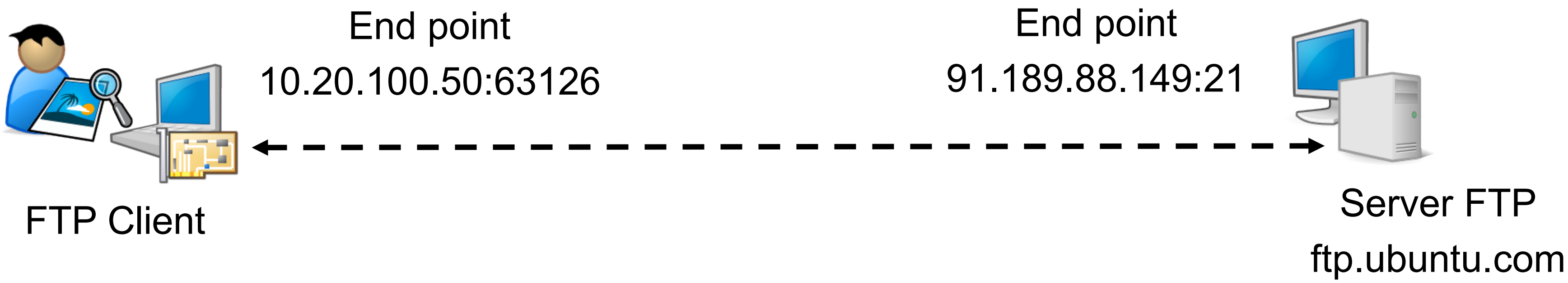
Session and connection

- At application level, 2 entities communicate using a session dialog
- A single session can include several transport connections at the same time
- FTP leverage 2 connections:
  - Command connection
  - Data connection



FTP Session

“command” Connection



End point

10.20.100.50:63126

End point

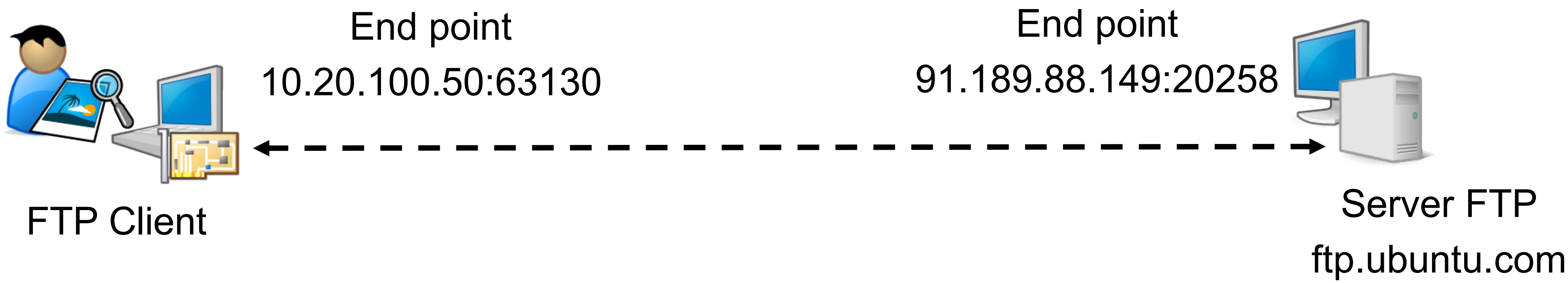
91.189.88.149:21

FTP Client

Server FTP

ftp.ubuntu.com

“data” Connection



End point

10.20.100.50:63130

End point

91.189.88.149:20258

FTP Client

Server FTP

ftp.ubuntu.com

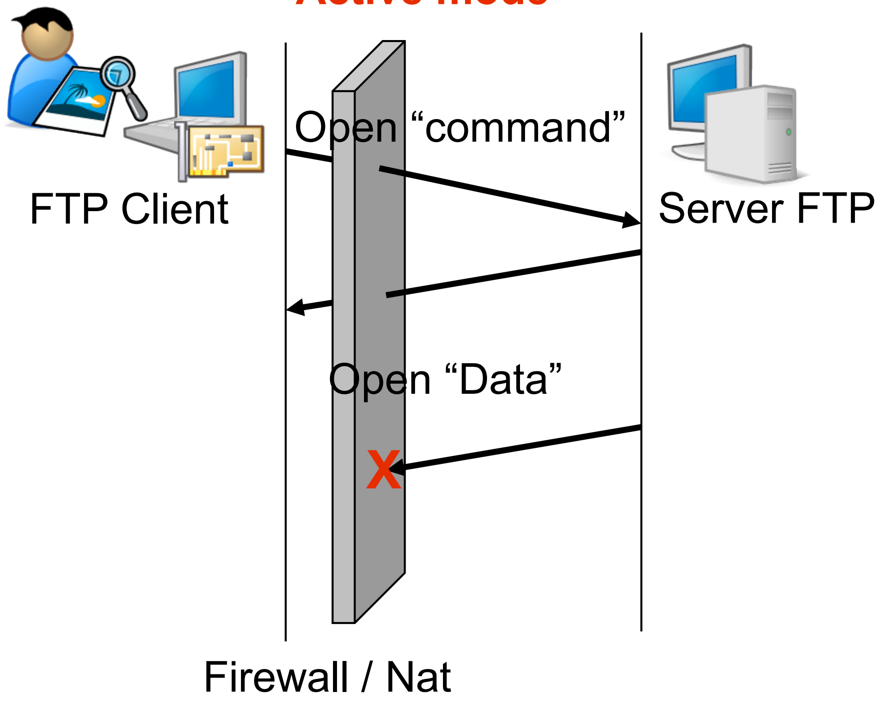
FTP: active mode and passive mode

- Who open the “command” connection?
  Clients connect to server on well known port 21
- Who open the “data” connection?
  - Active mode: server opens the “data” connection
  - Passive mode: client opens the “data” connection
- Passive mode
  - How does the client discover which port to use?
  - Port for “data” connection is provided by the server

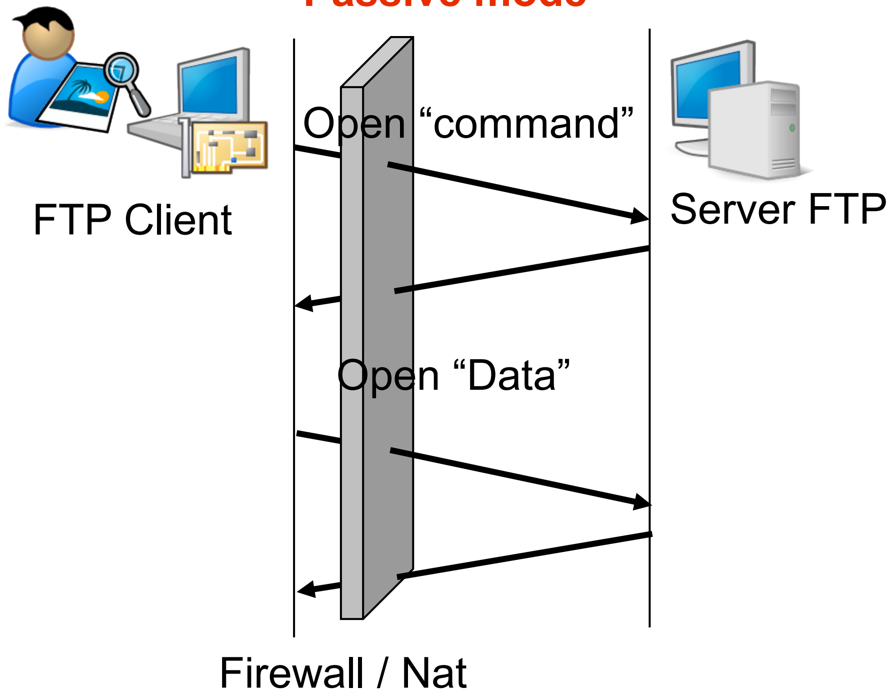
What is better, Active or Passive mode?

- Active mode is about sending a connection request from server to client
  - If the client is behind a NAT or a firewall, the connection request is blocked

Active mode



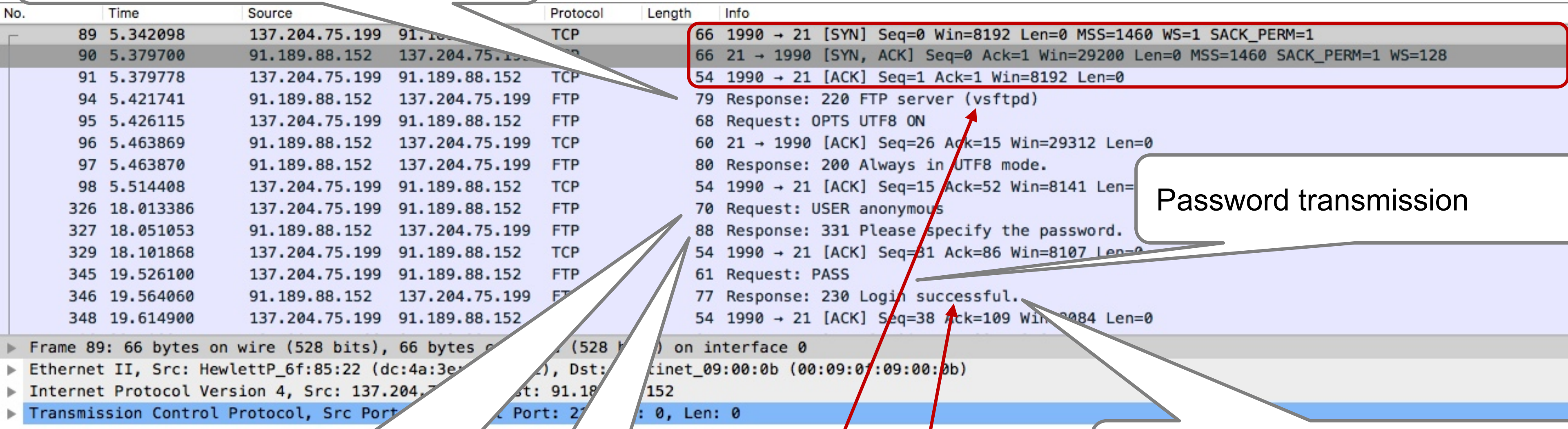
Passive mode



“command” connection opening

Server ftp Response

TCP three ways handshake



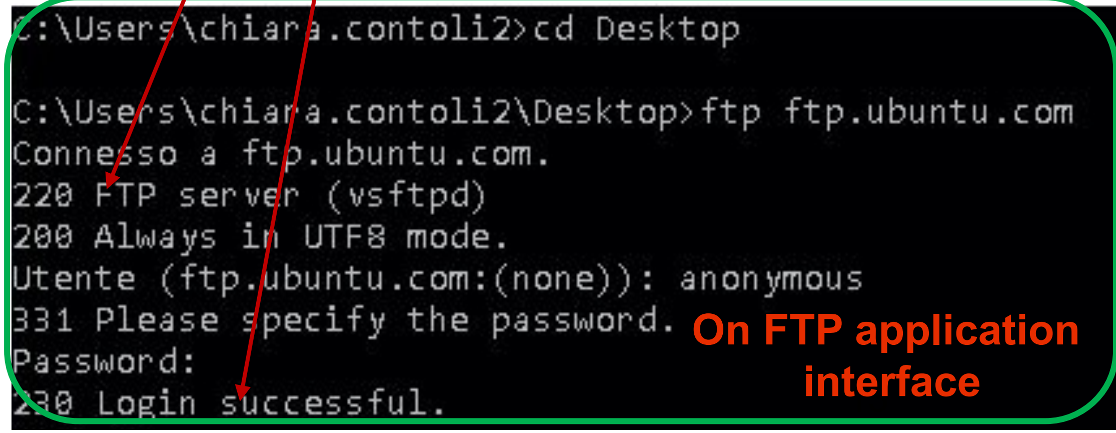
```
No.     Time        Source          Destination     Protocol  Length  Info
   89   5.342098    137.204.75.199  91.1            TCP               66  1990 → 21 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=1 SACK_PERM=1
   90   5.379700    91.189.88.152   137.204.75.1    P                 66  21 → 1990 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
   91   5.379778    137.204.75.199  91.189.88.152   TCP               54  1990 → 21 [ACK] Seq=1 Ack=1 Win=8192 Len=0
   94   5.421741    91.189.88.152   137.204.75.199  FTP               79  Response: 220 FTP server (vsftpd)
   95   5.426115    137.204.75.199  91.189.88.152   FTP               68  Request: OPTS UTF8 ON
   96   5.463869    91.189.88.152   137.204.75.199  TCP               60  21 → 1990 [ACK] Seq=26 Ack=15 Win=29312 Len=0
   97   5.463870    91.189.88.152   137.204.75.199  FTP               80  Response: 200 Always in UTF8 mode.
   98   5.514408    137.204.75.199  91.189.88.152   TCP               54  1990 → 21 [ACK] Seq=15 Ack=52 Win=8141 Len=
  326   18.013386   137.204.75.199  91.189.88.152   FTP               70  Request: USER anonymous
  327   18.051053   91.189.88.152   137.204.75.199  FTP               88  Response: 331 Please specify the password.
  329   18.101868   137.204.75.199  91.189.88.152   TCP               54  1990 → 21 [ACK] Seq=31 Ack=86 Win=8107 Len=0
  345   19.526100   137.204.75.199  91.189.88.152   FTP               61  Request: PASS
  346   19.564060   91.189.88.152   137.204.75.199  F                 77  Response: 230 Login successful.
  348   19.614900   137.204.75.199  91.189.88.152                     54  1990 → 21 [ACK] Seq=38 Ack=109 Win 084 Len=0
▶ Frame 89: 66 bytes on wire (528 bits), 66 bytes c         (528 b   ) on interface 0
▶ Ethernet II, Src: HewlettP_6f:85:22 (dc:4a:3e         ), Dst:    tinet_09:00:0b (00:09:0f:09:00:0b)
▶ Internet Protocol Version 4, Src: 137.204.7         st: 91.18    152
▶ Transmission Control Protocol, Src Port          Port: 21    : 0, Len: 0
```

Password transmission

Login successful

Connection request sent as “anonymous” user

Password request Response



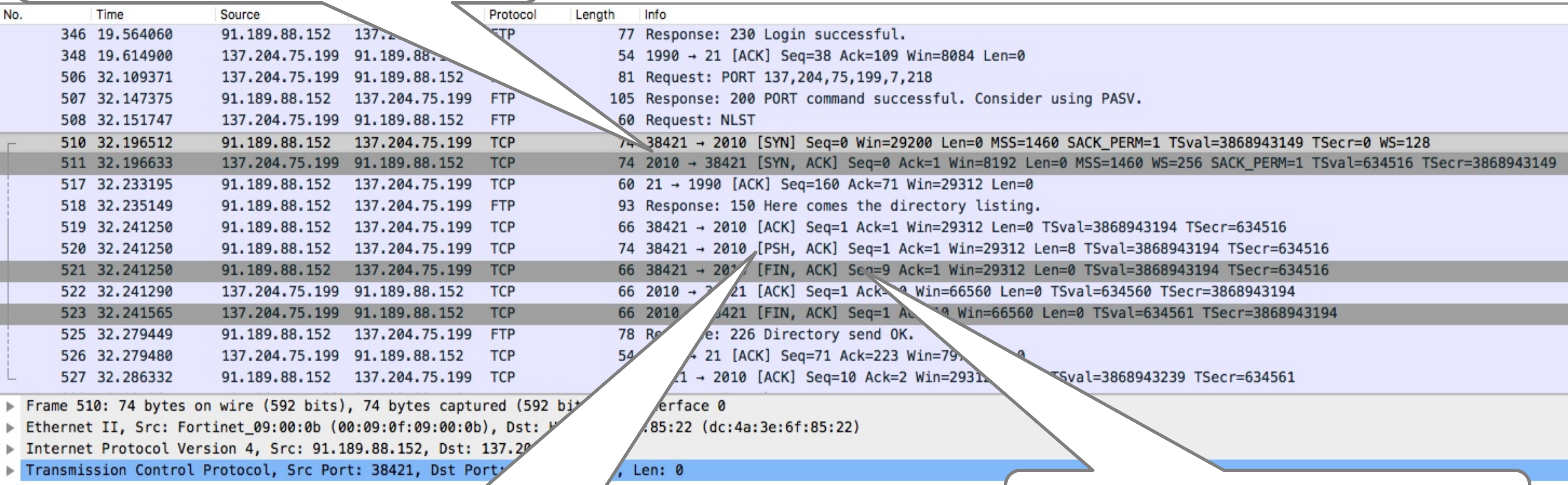
```
C:\Users\chiara.contoli2>cd Desktop

C:\Users\chiara.contoli2\Desktop>ftp ftp.ubuntu.com
Connesso a ftp.ubuntu.com.
220 FTP server (vsftpd)
200 Always in UTF8 mode.
Utente (ftp.ubuntu.com:(none)): anonymous
331 Please specify the password.
Password:
230 Login successful.
```

On FTP application interface

“data” connection

“data” connection opening

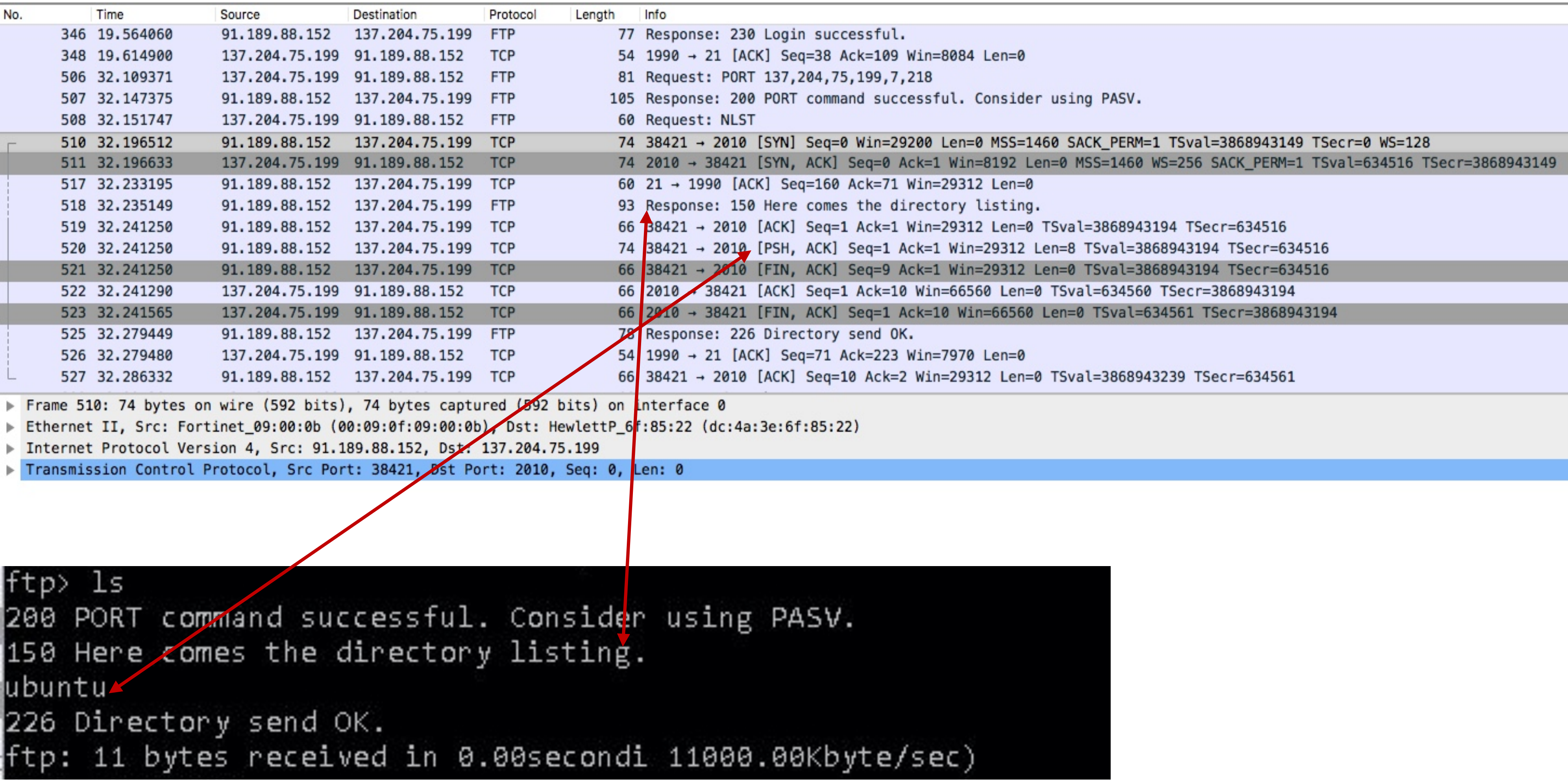


```
No.       Time        Source          Protocol    Length  Info
     346 19.564060   91.189.88.152   137.2              TP          77 Response: 230 Login successful.
     348 19.614900   137.204.75.199  91.189.88.                     54 1990 → 21 [ACK] Seq=38 Ack=109 Win=8084 Len=0
     506 32.109371   137.204.75.199  91.189.88.152                  81 Request: PORT 137,204,75,199,7,218
     507 32.147375   91.189.88.152   137.204.75.199  FTP           105 Response: 200 PORT command successful. Consider using PASV.
     508 32.151747   137.204.75.199  91.189.88.152   FTP            60 Request: NLST
     510 32.196512   91.189.88.152   137.204.75.199  TCP            74 38421 → 2010 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3868943149 TSecr=0 WS=128
     511 32.196633   137.204.75.199  91.189.88.152   TCP            74 2010 → 38421 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=634516 TSecr=3868943149
     517 32.233195   91.189.88.152   137.204.75.199  TCP            60 21 → 1990 [ACK] Seq=160 Ack=71 Win=29312 Len=0
     518 32.235149   91.189.88.152   137.204.75.199  FTP            93 Response: 150 Here comes the directory listing.
     519 32.241250   91.189.88.152   137.204.75.199  TCP            66 38421 → 2010 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=3868943194 TSecr=634516
     520 32.241250   91.189.88.152   137.204.75.199  TCP            74 38421 → 2010 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=8 TSval=3868943194 TSecr=634516
     521 32.241250   91.189.88.152   137.204.75.199  TCP            66 38421 → 201  [FIN, ACK] Se =9 Ack=1 Win=29312 Len=0 TSval=3868943194 TSecr=634516
     522 32.241290   137.204.75.199  91.189.88.152   TCP            66 2010 →   21 [ACK] Seq=1 Ack   Win=66560 Len=0 TSval=634560 TSecr=3868943194
     523 32.241565   137.204.75.199  91.189.88.152   TCP            66 2010   421 [FIN, ACK] Seq=1 A    Win=66560 Len=0 TSval=634561 TSecr=3868943194
     525 32.279449   91.189.88.152   137.204.75.199  FTP            78 Re    e: 226 Directory send OK.
     526 32.279480   137.204.75.199  91.189.88.152   TCP            54  → 21 [ACK] Seq=71 Ack=223 Win=79      0
     527 32.286332   91.189.88.152   137.204.75.199  TCP               1 → 2010 [ACK] Seq=10 Ack=2 Win=2931       TSval=3868943239 TSecr=634561
▶ Frame 510: 74 bytes on wire (592 bits), 74 bytes captured (592 bi         erface 0
▶ Ethernet II, Src: Fortinet_09:00:0b (00:09:0f:09:00:0b), Dst: H      :85:22 (dc:4a:3e:6f:85:22)
▶ Internet Protocol Version 4, Src: 91.189.88.152, Dst: 137.20
▶ Transmission Control Protocol, Src Port: 38421, Dst Port:          , Len: 0
```

“data” connection closing

Sending data on “data” connection

Sending data on “data” connection

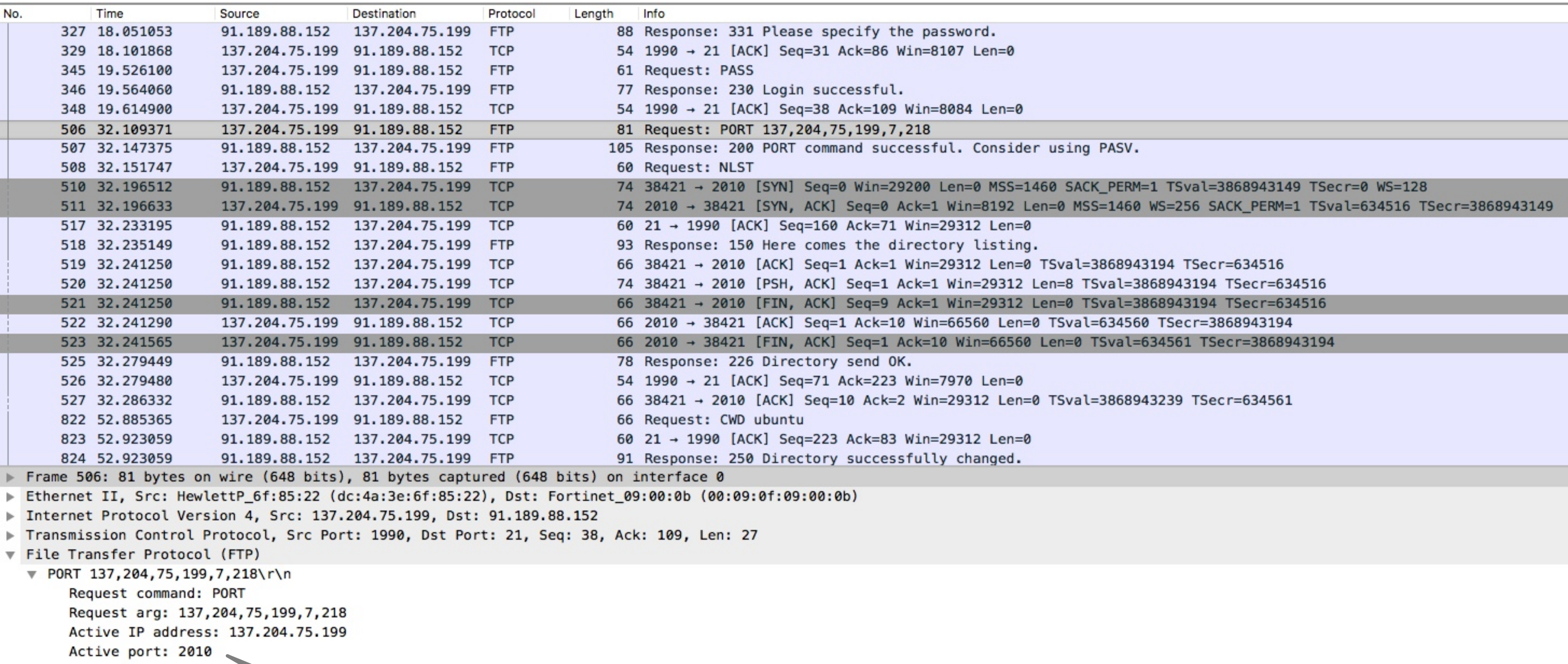


```
No.  Time       Source          Destination     Protocol  Length  Info
346  19.564060  91.189.88.152   137.204.75.199  FTP           77  Response: 230 Login successful.
348  19.614900  137.204.75.199  91.189.88.152   TCP           54  1990 → 21 [ACK] Seq=38 Ack=109 Win=8084 Len=0
506  32.109371  137.204.75.199  91.189.88.152   FTP           81  Request: PORT 137,204,75,199,7,218
507  32.147375  91.189.88.152   137.204.75.199  FTP          105  Response: 200 PORT command successful. Consider using PASV.
508  32.151747  137.204.75.199  91.189.88.152   FTP           60  Request: NLST
510  32.196512  91.189.88.152   137.204.75.199  TCP           74  38421 → 2010 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3868943149 TSecr=0 WS=128
511  32.196633  137.204.75.199  91.189.88.152   TCP           74  2010 → 38421 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=634516 TSecr=3868943149
517  32.233195  91.189.88.152   137.204.75.199  TCP           60  21 → 1990 [ACK] Seq=160 Ack=71 Win=29312 Len=0
518  32.235149  91.189.88.152   137.204.75.199  FTP           93  Response: 150 Here comes the directory listing.
519  32.241250  91.189.88.152   137.204.75.199  TCP           66  38421 → 2010 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=3868943194 TSecr=634516
520  32.241250  91.189.88.152   137.204.75.199  TCP           74  38421 → 2010 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=8 TSval=3868943194 TSecr=634516
521  32.241250  91.189.88.152   137.204.75.199  TCP           66  38421 → 2010 [FIN, ACK] Seq=9 Ack=1 Win=29312 Len=0 TSval=3868943194 TSecr=634516
522  32.241290  137.204.75.199  91.189.88.152   TCP           66  2010 → 38421 [ACK] Seq=1 Ack=10 Win=66560 Len=0 TSval=634560 TSecr=3868943194
523  32.241565  137.204.75.199  91.189.88.152   TCP           66  2010 → 38421 [FIN, ACK] Seq=1 Ack=10 Win=66560 Len=0 TSval=634561 TSecr=3868943194
525  32.279449  91.189.88.152   137.204.75.199  FTP           78  Response: 226 Directory send OK.
526  32.279480  137.204.75.199  91.189.88.152   TCP           54  1990 → 21 [ACK] Seq=71 Ack=223 Win=7970 Len=0
527  32.286332  91.189.88.152   137.204.75.199  TCP           66  38421 → 2010 [ACK] Seq=10 Ack=2 Win=29312 Len=0 TSval=3868943239 TSecr=634561

▶ Frame 510: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
▶ Ethernet II, Src: Fortinet_09:00:0b (00:09:0f:09:00:0b), Dst: HewlettP_6f:85:22 (dc:4a:3e:6f:85:22)
▶ Internet Protocol Version 4, Src: 91.189.88.152, Dst: 137.204.75.199
▶ Transmission Control Protocol, Src Port: 38421, Dst Port: 2010, Seq: 0, Len: 0
```

```
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
ubuntu
226 Directory send OK.
ftp: 11 bytes received in 0.00secondi 11000.00Kbyte/sec)
```

Active mode requested by client



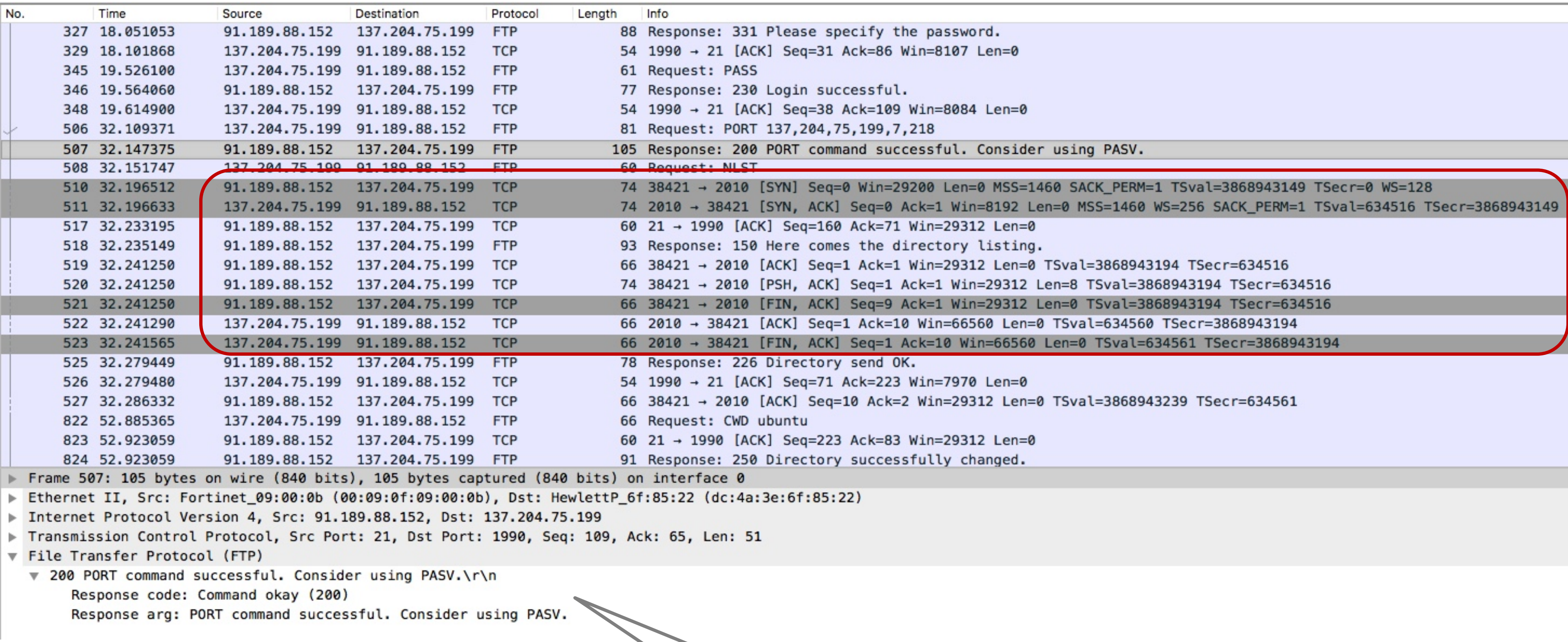
```
No.   Time        Source          Destination     Protocol  Length  Info
  327 18.051053   91.189.88.152   137.204.75.199  FTP           88  Response: 331 Please specify the password.
  329 18.101868   137.204.75.199  91.189.88.152   TCP           54  1990 → 21 [ACK] Seq=31 Ack=86 Win=8107 Len=0
  345 19.526100   137.204.75.199  91.189.88.152   FTP           61  Request: PASS
  346 19.564060   91.189.88.152   137.204.75.199  FTP           77  Response: 230 Login successful.
  348 19.614900   137.204.75.199  91.189.88.152   TCP           54  1990 → 21 [ACK] Seq=38 Ack=109 Win=8084 Len=0
  506 32.109371   137.204.75.199  91.189.88.152   FTP           81  Request: PORT 137,204,75,199,7,218
  507 32.147375   91.189.88.152   137.204.75.199  FTP          105  Response: 200 PORT command successful. Consider using PASV.
  508 32.151747   137.204.75.199  91.189.88.152   FTP           60  Request: NLST
  510 32.196512   91.189.88.152   137.204.75.199  TCP           74  38421 → 2010 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3868943149 TSecr=0 WS=128
  511 32.196633   137.204.75.199  91.189.88.152   TCP           74  2010 → 38421 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=634516 TSecr=3868943149
  517 32.233195   91.189.88.152   137.204.75.199  TCP           60  21 → 1990 [ACK] Seq=160 Ack=71 Win=29312 Len=0
  518 32.235149   91.189.88.152   137.204.75.199  FTP           93  Response: 150 Here comes the directory listing.
  519 32.241250   91.189.88.152   137.204.75.199  TCP           66  38421 → 2010 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=3868943194 TSecr=634516
  520 32.241250   91.189.88.152   137.204.75.199  TCP           74  38421 → 2010 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=8 TSval=3868943194 TSecr=634516
  521 32.241250   91.189.88.152   137.204.75.199  TCP           66  38421 → 2010 [FIN, ACK] Seq=9 Ack=1 Win=29312 Len=0 TSval=3868943194 TSecr=634516
  522 32.241290   137.204.75.199  91.189.88.152   TCP           66  2010 → 38421 [ACK] Seq=1 Ack=10 Win=66560 Len=0 TSval=634560 TSecr=3868943194
  523 32.241565   137.204.75.199  91.189.88.152   TCP           66  2010 → 38421 [FIN, ACK] Seq=1 Ack=10 Win=66560 Len=0 TSval=634561 TSecr=3868943194
  525 32.279449   91.189.88.152   137.204.75.199  FTP           78  Response: 226 Directory send OK.
  526 32.279480   137.204.75.199  91.189.88.152   TCP           54  1990 → 21 [ACK] Seq=71 Ack=223 Win=7970 Len=0
  527 32.286332   91.189.88.152   137.204.75.199  TCP           66  38421 → 2010 [ACK] Seq=10 Ack=2 Win=29312 Len=0 TSval=3868943239 TSecr=634561
  822 52.885365   137.204.75.199  91.189.88.152   FTP           66  Request: CWD ubuntu
  823 52.923059   91.189.88.152   137.204.75.199  TCP           60  21 → 1990 [ACK] Seq=223 Ack=83 Win=29312 Len=0
  824 52.923059   91.189.88.152   137.204.75.199  FTP           91  Response: 250 Directory successfully changed.
```

```
▶ Frame 506: 81 bytes on wire (648 bits), 81 bytes captured (648 bits) on interface 0
▶ Ethernet II, Src: HewlettP_6f:85:22 (dc:4a:3e:6f:85:22), Dst: Fortinet_09:00:0b (00:09:0f:09:00:0b)
▶ Internet Protocol Version 4, Src: 137.204.75.199, Dst: 91.189.88.152
▶ Transmission Control Protocol, Src Port: 1990, Dst Port: 21, Seq: 38, Ack: 109, Len: 27
▼ File Transfer Protocol (FTP)
  ▼ PORT 137,204,75,199,7,218\r\n
      Request command: PORT
      Request arg: 137,204,75,199,7,218
      Active IP address: 137.204.75.199
      Active port: 2010
```

Client requests Active Mode (PORT)

PORT is an option offered by server

Server opens a connection

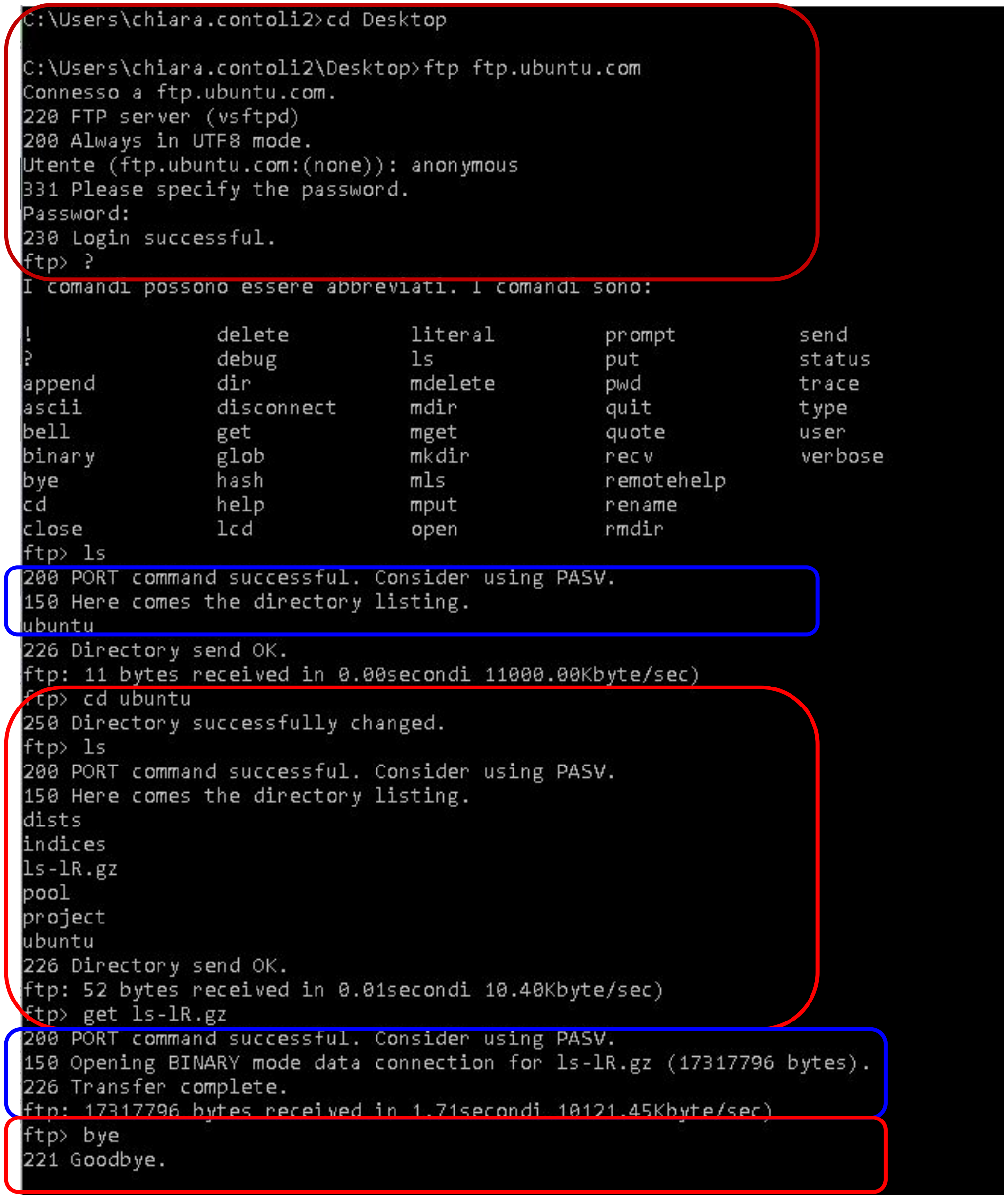


No.	Time	Source	Destination	Protocol	Length	Info
327	18.051053	91.189.88.152	137.204.75.199	FTP	88	Response: 331 Please specify the password.
329	18.101868	137.204.75.199	91.189.88.152	TCP	54	1990 → 21 [ACK] Seq=31 Ack=86 Win=8107 Len=0
345	19.526100	137.204.75.199	91.189.88.152	FTP	61	Request: PASS
346	19.564060	91.189.88.152	137.204.75.199	FTP	77	Response: 230 Login successful.
348	19.614900	137.204.75.199	91.189.88.152	TCP	54	1990 → 21 [ACK] Seq=38 Ack=109 Win=8084 Len=0
506	32.109371	137.204.75.199	91.189.88.152	FTP	81	Request: PORT 137,204,75,199,7,218
507	32.147375	91.189.88.152	137.204.75.199	FTP	105	Response: 200 PORT command successful. Consider using PASV.
508	32.151747	137.204.75.199	91.189.88.152	FTP	60	Request: NLST
510	32.196512	91.189.88.152	137.204.75.199	TCP	74	38421 → 2010 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3868943149 TSecr=0 WS=128
511	32.196633	137.204.75.199	91.189.88.152	TCP	74	2010 → 38421 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=634516 TSecr=3868943149
517	32.233195	91.189.88.152	137.204.75.199	TCP	60	21 → 1990 [ACK] Seq=160 Ack=71 Win=29312 Len=0
518	32.235149	91.189.88.152	137.204.75.199	FTP	93	Response: 150 Here comes the directory listing.
519	32.241250	91.189.88.152	137.204.75.199	TCP	66	38421 → 2010 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=3868943194 TSecr=634516
520	32.241250	91.189.88.152	137.204.75.199	TCP	74	38421 → 2010 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=8 TSval=3868943194 TSecr=634516
521	32.241250	91.189.88.152	137.204.75.199	TCP	66	38421 → 2010 [FIN, ACK] Seq=9 Ack=1 Win=29312 Len=0 TSval=3868943194 TSecr=634516
522	32.241290	137.204.75.199	91.189.88.152	TCP	66	2010 → 38421 [ACK] Seq=1 Ack=10 Win=66560 Len=0 TSval=634560 TSecr=3868943194
523	32.241565	137.204.75.199	91.189.88.152	TCP	66	2010 → 38421 [FIN, ACK] Seq=1 Ack=10 Win=66560 Len=0 TSval=634561 TSecr=3868943194
525	32.279449	91.189.88.152	137.204.75.199	FTP	78	Response: 226 Directory send OK.
526	32.279480	137.204.75.199	91.189.88.152	TCP	54	1990 → 21 [ACK] Seq=71 Ack=223 Win=7970 Len=0
527	32.286332	91.189.88.152	137.204.75.199	TCP	66	38421 → 2010 [ACK] Seq=10 Ack=2 Win=29312 Len=0 TSval=3868943239 TSecr=634561
822	52.885365	137.204.75.199	91.189.88.152	FTP	66	Request: CWD ubuntu
823	52.923059	91.189.88.152	137.204.75.199	TCP	60	21 → 1990 [ACK] Seq=223 Ack=83 Win=29312 Len=0
824	52.923059	91.189.88.152	137.204.75.199	FTP	91	Response: 250 Directory successfully changed.

```
▶ Frame 507: 105 bytes on wire (840 bits), 105 bytes captured (840 bits) on interface 0
▶ Ethernet II, Src: Fortinet_09:00:0b (00:09:0f:09:00:0b), Dst: HewlettP_6f:85:22 (dc:4a:3e:6f:85:22)
▶ Internet Protocol Version 4, Src: 91.189.88.152, Dst: 137.204.75.199
▶ Transmission Control Protocol, Src Port: 21, Dst Port: 1990, Seq: 109, Ack: 65, Len: 51
▼ File Transfer Protocol (FTP)
  ▼ 200 PORT command successful. Consider using PASV.\r\n
      Response code: Command okay (200)
      Response arg: PORT command successful. Consider using PASV.
```

Server Response contains the positive reply to open “data” connection in active mode

FTP session: application



```
C:\Users\chiara.contoli2>cd Desktop

C:\Users\chiara.contoli2\Desktop>ftp ftp.ubuntu.com
Connesso a ftp.ubuntu.com.
220 FTP server (vsftpd)
200 Always in UTF8 mode.
Utente (ftp.ubuntu.com:(none)): anonymous
331 Please specify the password.
Password:
230 Login successful.
ftp> ?
I comandi possono essere abbreviati. I comandi sono:

!               delete          literal         prompt          send
?               debug           ls              put             status
append          dir             mdelete         pwd             trace
ascii           disconnect      mdir            quit            type
bell            get             mget            quote           user
binary          glob            mkdir           recv            verbose
bye             hash            mls             remotehelp
cd              help            mput            rename
close           lcd             open            rmdir
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
ubuntu
226 Directory send OK.
ftp: 11 bytes received in 0.00secondi 11000.00Kbyte/sec)
ftp> cd ubuntu
250 Directory successfully changed.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
dists
indices
ls-lR.gz
pool
project
ubuntu
226 Directory send OK.
ftp: 52 bytes received in 0.01secondi 10.40Kbyte/sec)
ftp> get ls-lR.gz
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for ls-lR.gz (17317796 bytes).
226 Transfer complete.
ftp: 17317796 bytes received in 1.71secondi 10121.45Kbyte/sec)
ftp> bye
221 Goodbye.
```

"command" connection

Client Port : 1990

Server Port : 21

"data" connection

Client Port : 2010

Server Port : 38421

"command" connection

"data" connection

Client Port : 2070

Server Port : 36121

"command" connection

Few words about Telnet

- Application level protocol (over Transmission Control Protocol, TCP) that provides
  - Bi-directional communication
  - Data transfer
  - Remote connection
- Telnet adopts well known port 25

Names and network addresses: Domain Name System

Programmazione di Reti

Ing. Chiara Contoli, PhD
chiara.contoli@unibo.it

Corso di Laurea Triennale in
Ingegneria e Scienze Informatiche

Names and addresses

- For user convenience, IP addresses are associated to symbolic names
- Symbolic name
  - Alphanumeric strings splitted by dots

    http://www.informatica.unibo.it/
  - String names are virtually infinite

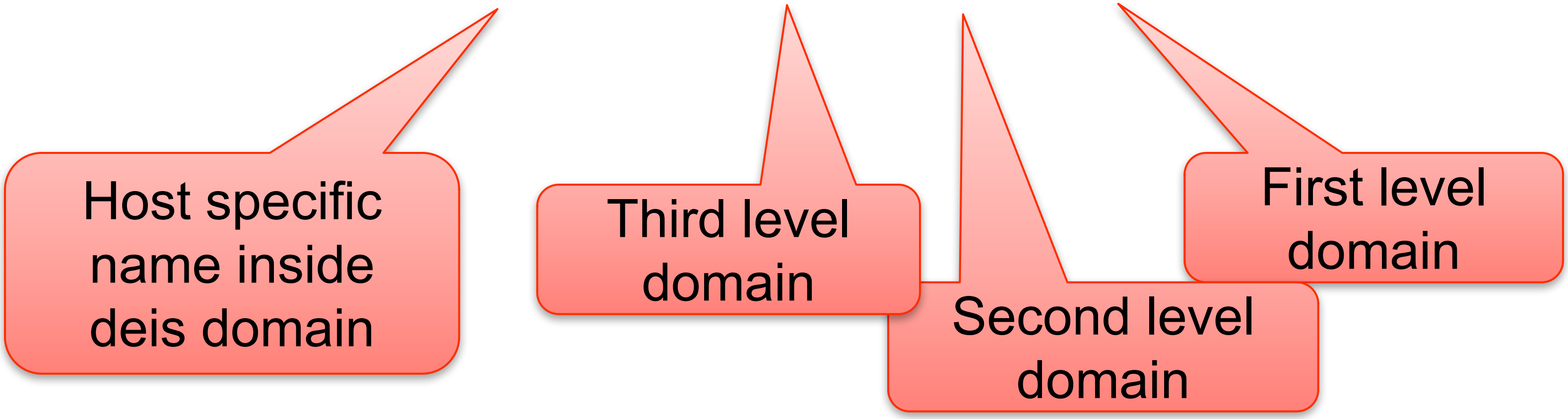
Which is the name composition?

- Strings are not arbitrarily chosen
- Name composition reflects a hierarchical Domain organization
- Domains are associated with conventional names
  - it = string identifying Italy domain
  - unibo = string identifying University of Bologna domain
  - informatica = string identifying a Department inside Unibo
- Domain can be splitted in subdomain
  - *unibo* is a subdomain of *it*
  - *Informatica* might have subdomain as well

The name

- Names sequence start from the right most part

deisnet.deis.unibo.it



Host specific name inside deis domain

Third level domain

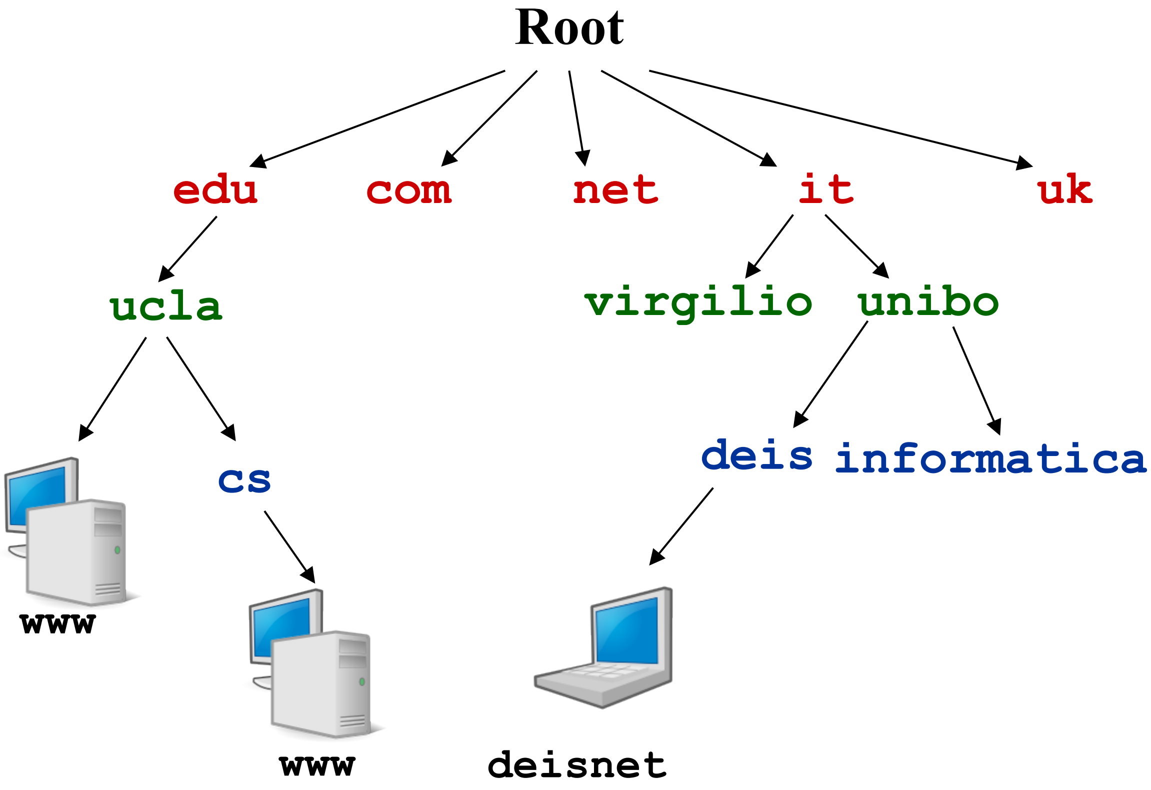
Second level domain

First level domain

- Host specific name is arbitrary
- Domain names are assigned by IANA

Example

deisnet.deis.unibo.it



Root

edu com net it uk

ucla

virgilio unibo

www cs

deis informatica

www

deisnet

PRIMARY DOMAINS

SECOND LEVEL DOMAINS

THIRD LEVEL DOMAINS

HOST

PRIMARY DOMAINS

edu	educational and research organization in USA
gov	governmental organization in USA
com	commercial organization
mil	military groups in USA
org	other organizations
net	centers for network support

country code standard acronym for nation identification (ISO 3166)

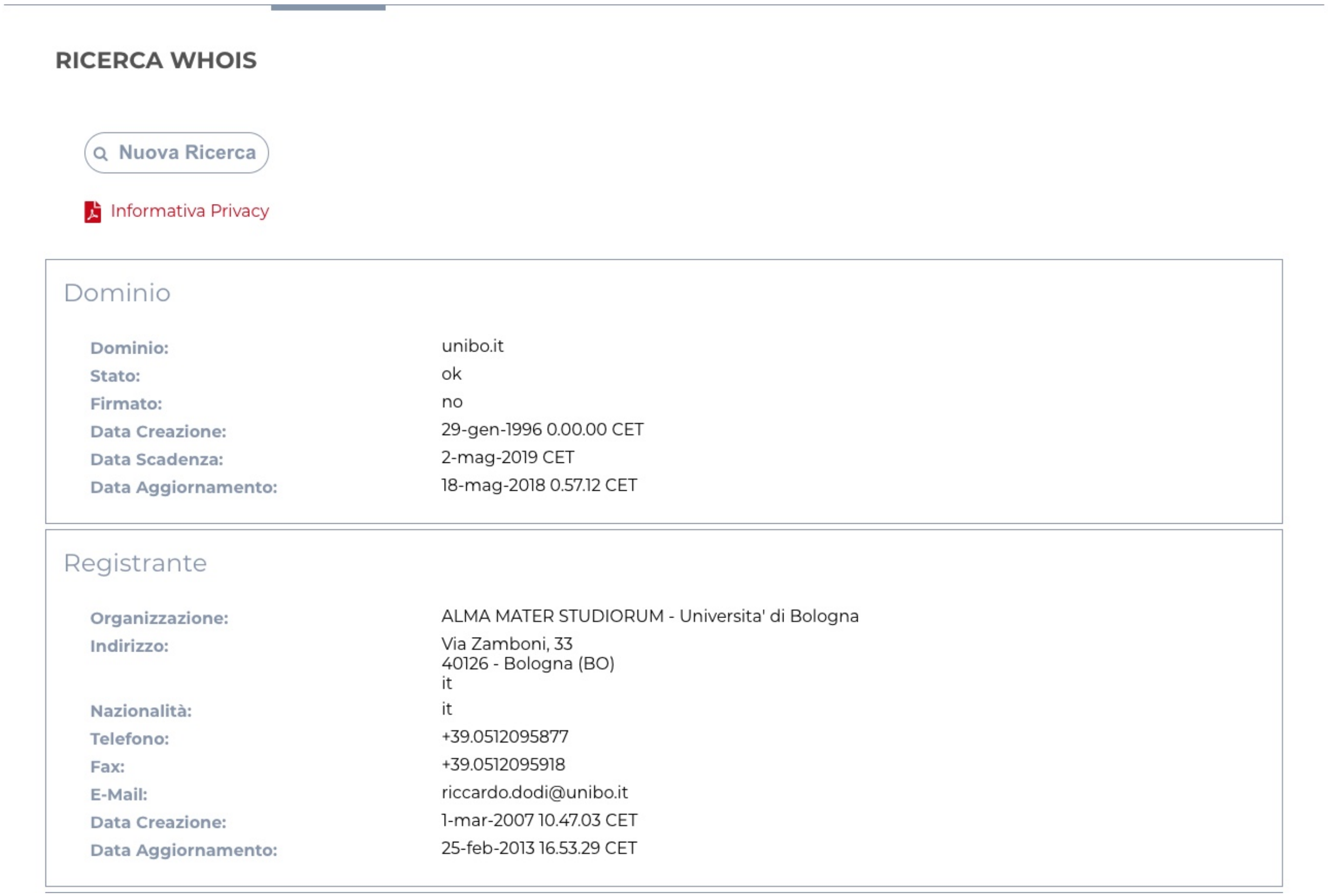
it fr uk de au jp ie dk br …

Registro.it (www.nic.it)

- Registro is a sort of “civil registry” for *.it* Internet domains
  - Only here you are allowed to ask for, modify or remove one or more *.it* domains
- Upon users request, Registro associates a numerical addresses group to a name
  - Such relation is memorized on the Dbna (database of assigned names). Dbna needs to be reachable from each computer on the Internet in order to be able to connect to a *.it* domain
- Rules on global network are established by an international organization known as ICANN (Internet Corporation for Assigned Names and Numbers).
- In 1987, ICANN designated the National Council of Research to be *.it* Internet domain manager
  - Registro.it was born this way, located at the Computer Science and Telematics Institute of Cnr in Pisa

Whois service

- Whois service allows to verify if and to who a certain domain is assigned or not
- Searching for unibo.it returns the following



RICERCA WHOIS

Dominio

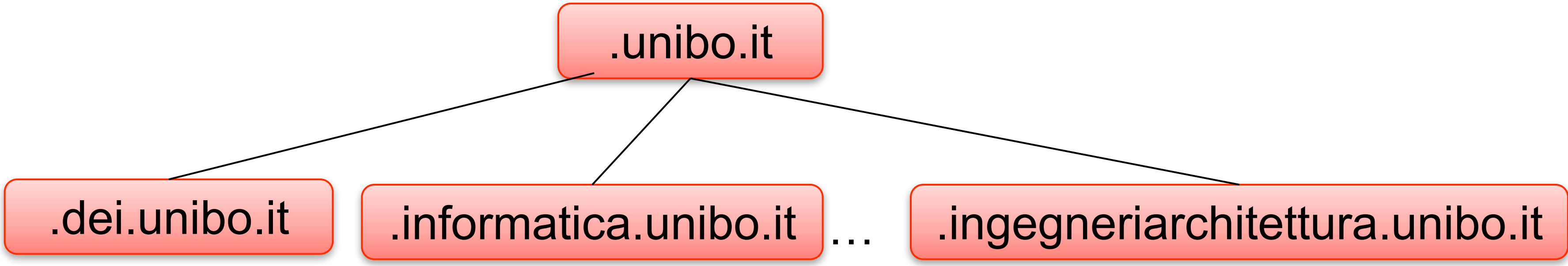
Dominio:	unibo.it
Stato:	ok
Firmato:	no
Data Creazione:	29-gen-1996 0.00.00 CET
Data Scadenza:	2-mag-2019 CET
Data Aggiornamento:	18-mag-2018 0.57.12 CET

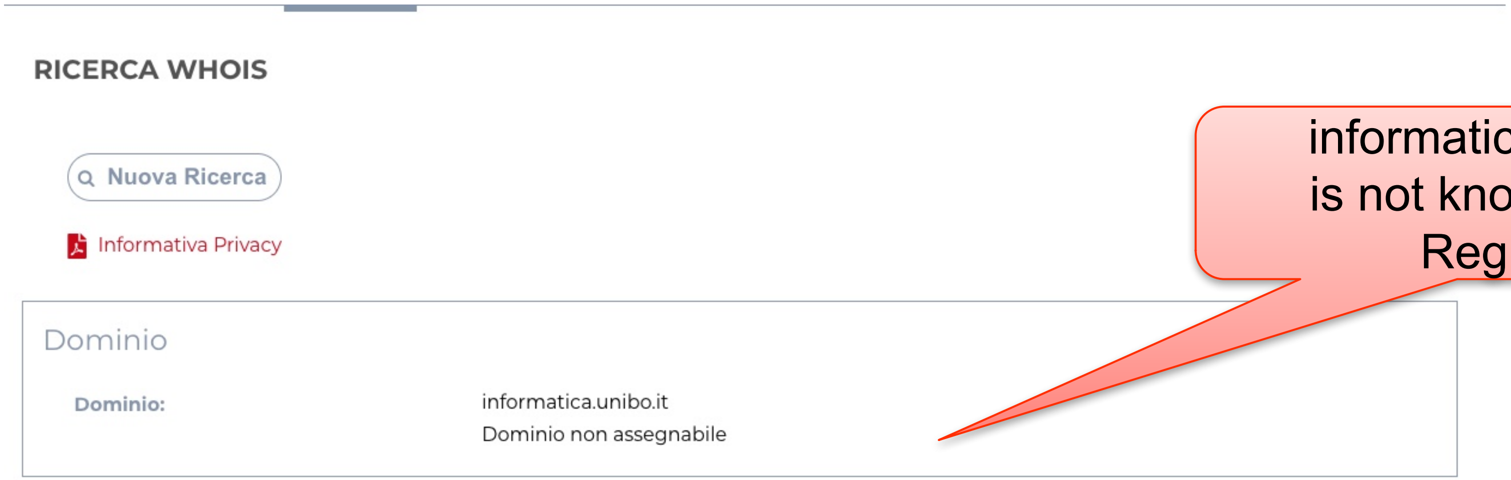
Registrante

Organizzazione:	ALMA MATER STUDIORUM - Universita' di Bologna
Indirizzo:	Via Zamboni, 33 40126 - Bologna (BO) it
Nazionalità:	it
Telefono:	+39.0512095877
Fax:	+39.0512095918
E-Mail:	riccardo.dodi@unibo.it
Data Creazione:	1-mar-2007 10.47.03 CET
Data Aggiornamento:	25-feb-2013 16.53.29 CET

The hierarchy

- The recipient of the domain is responsible for possible subdomain management
  - Subdomains are not registered





RICERCA WHOIS

Dominio

Dominio: informatica.unibo.it
Dominio non assegnabile

informatica.unibo.it is not known at the Registro

Domain Name System

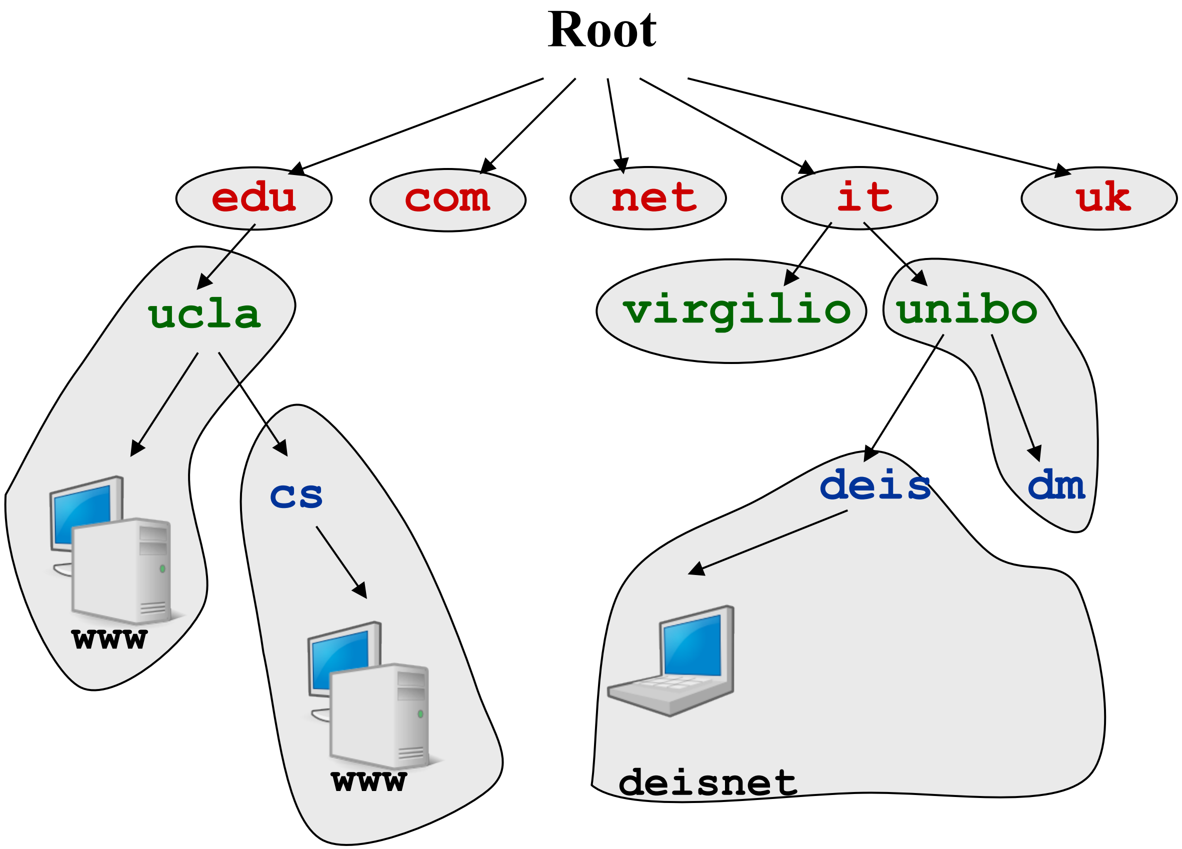
- An automatic service is adopted to resolve IP addresses starting from symbolic names
  - It's a sort of digital telephone list
- Domain Name System (DNS) is a distributed database that link to each Name a corresponding network address
- DNS "check" is performed through proper DNS "server"
  - Such check is transparent to the user
    - Browser knows how to interact with the DNS without interacting with the end user

Names database management

- **PROBLEM** – how do we handle a database containing all hosts Internet names?

- **SOLUTION** – distributed database
  - Names space is divided in non overlapping **zone**, which contain one or more subdomains
  - Each zone is composed of a main **name server** and one or more secondary servers
  - Each name server knows about IP addresses corresponding to hosts contained in its zone, for which the name server is responsible for

Zone subdivision



PRIMARY DOMAIN

SECOND LEVEL DOMAIN

THIRD LEVEL DOMAIN

HOST

Resolve a name

- In order to resolve a name to an IP address
  - Hosts need to be equipped with a specific service known as name resolver
    - It depends on implementation and operating system
  - Host needs to be configured with IP address(es) of DNS server(s) of the belonging zone
  - Hosts can be pre-configured with some links names-addresses in a local archive
    - File name and syntax are implementation dependent
  - When an application needs to solve a name, it calls the *name resolver*

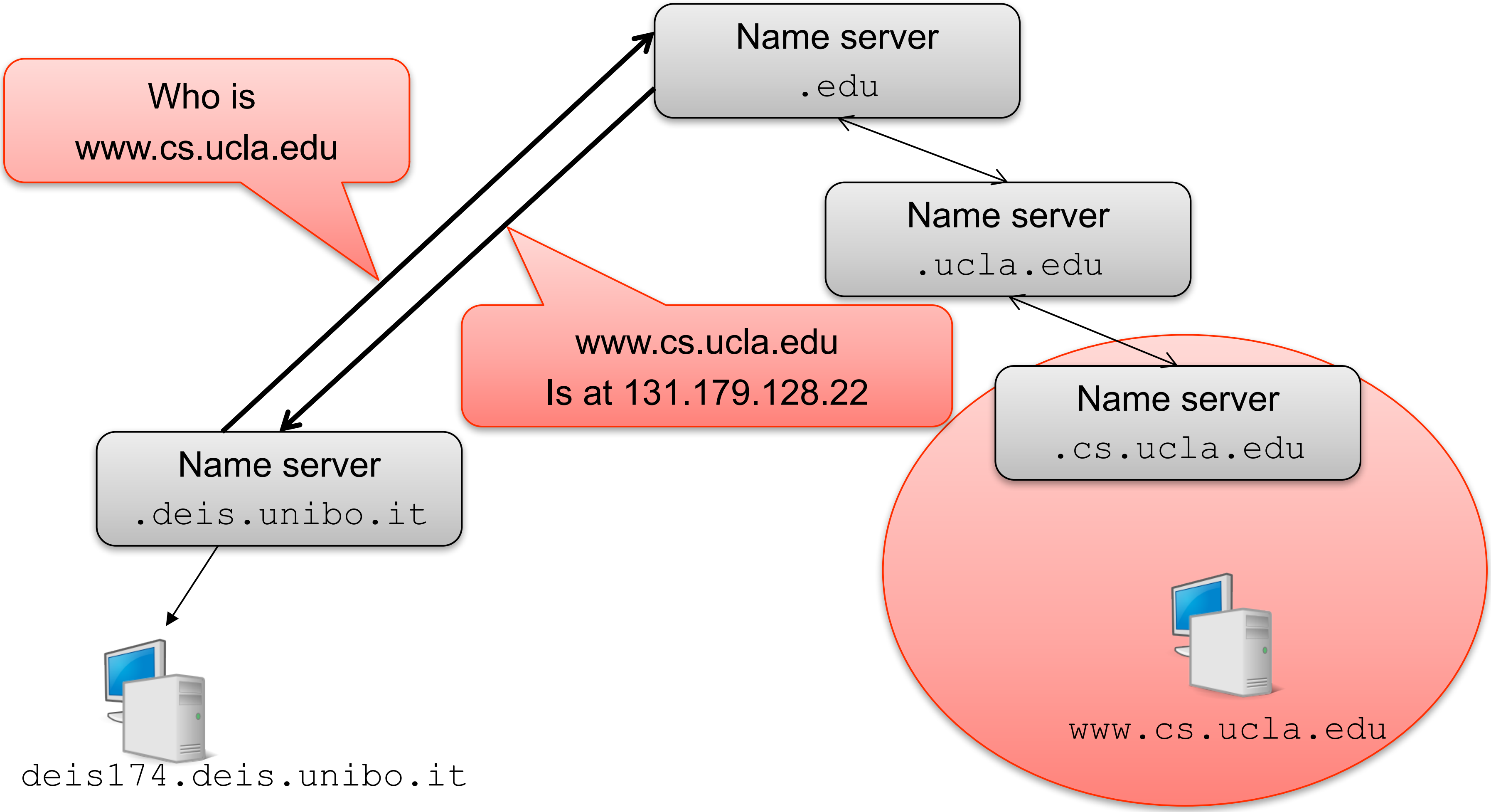
Name resolver

- Different situations may occur:
  - Name resolver may resolve the name locally (thanks to a local archive, cache or file)
    - It communicates directly with application IP address
  - Name resolver may not resolve the name locally
    - It sends a query to the name server of the zone to which the host belongs to
- Name server of the zone resolve the name cooperating with DNS servers of other zones
  - First of all, it contacts the name server of the primary level domain
  - Possibly, it contacts sublevel domains

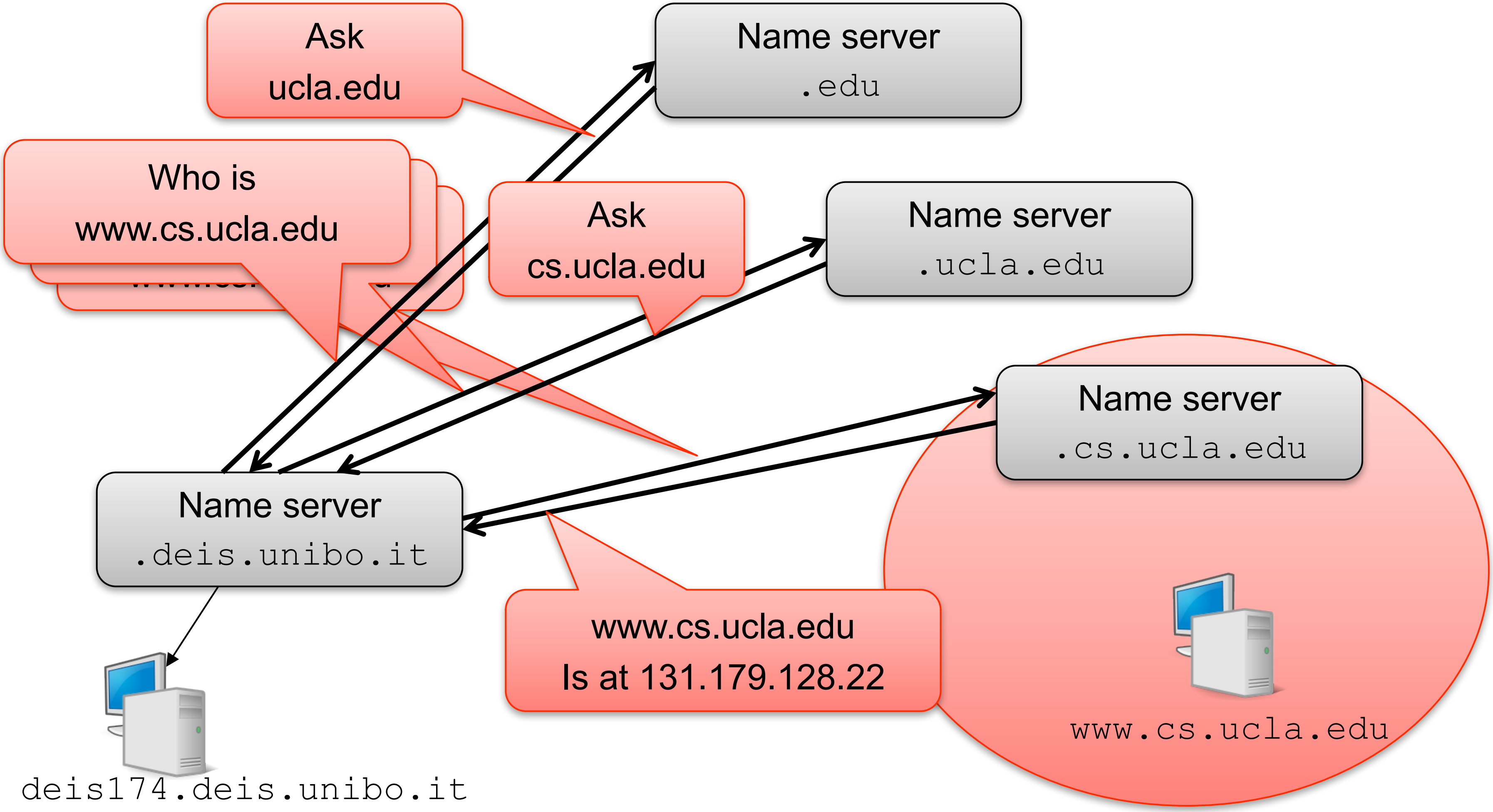
Iterative and recursive response

- Response to the query to the zone name server can be
  - Recursive
    - Queried name server is responsible for resolving the name possibly by sending a query to subdomain servers and then by sending back the response
  - Iterative
    - Queried name server send back a response indicating a subdomain name server to which the request will be forwarded in order to be resolved

Recursive mode



Iterative mode



Example (1)

- An application running on **`deis174.deis.unibo.it`** needs to contact host **`deisnet.deis.unibo.it`** of which it does not know the IP address
- To get the address, it calls a local application known as **name resolver**
- If the resolver already has the requested information (e.g. in cache or file) the response is directly communicated to the application
- Otherwise, the resolver queries the zone name server to which the local host belongs to, that is **`deis.unibo.it`** name server
- This name server has the information (because belongs to its competence) and it sends back the response to the resolver

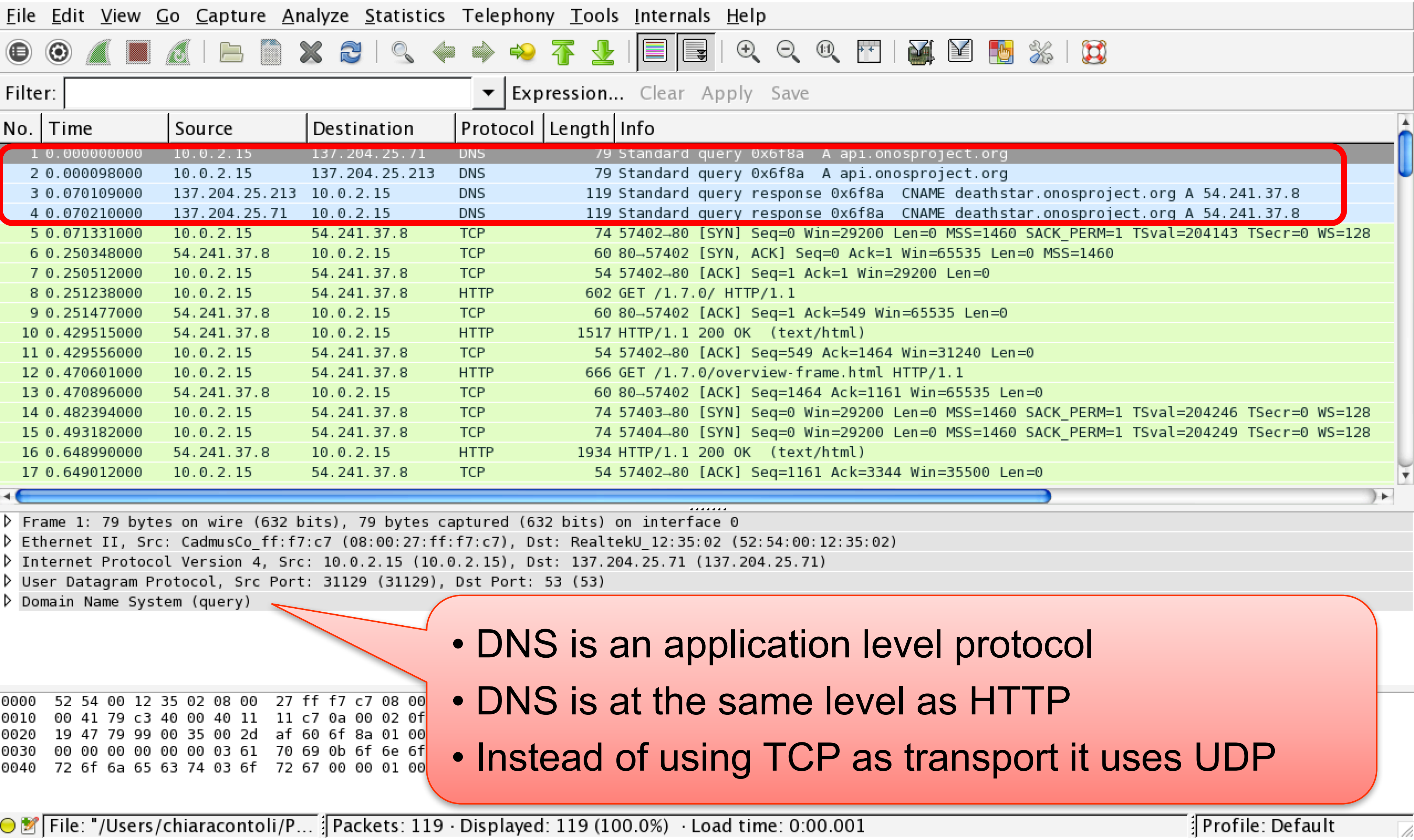
Example (2)

- An application running on **`deis174.deis.unibo.it`** needs to contact host **`www.cs.ucla.edu`** of which it does not know the IP address
- To get the address, it calls a local application known as name resolver
- If the resolver already has the requested information (e.g. in cache or file) response it's directly communicated to the application
- Otherwise, the resolver queries the zone name server to which the local host belongs to, that is **`deis.unibo.it`** name server
- If this name server has already the information (because in cache) it sends back the response to the resolver

Example (3)

- Otherwise, it contacts the name server of the primary level domain of the requested host, that is **`.edu`** (if it does not know the address, it asks for this information to one of the so called **root-server**)
- **`.edu`** name server provide name server address of **`.ucla.edu`** that, if does not know the required information, it send back the request to **`.cs.ucla.edu`** name server adopting two possible mode:
  - **recursive**: **`.ucla.edu`** name server search for the information and then send back the response to **`.deis.unibo.it`** name server
  - **Iterative**: **`.deis.unibo.it`** name server queries **`.cs.ucla.edu`** name server
- **`.deis.unibo.it`** name server sends to the resolver the required address

DNS Request


File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save

```
No. Time         Source         Destination    Protocol Length Info
  1 0.000000000  10.0.2.15      137.204.25.71  DNS          79 Standard query 0x6f8a  A api.onosproject.org
  2 0.000098000  10.0.2.15      137.204.25.213 DNS          79 Standard query 0x6f8a  A api.onosproject.org
  3 0.070109000  137.204.25.213 10.0.2.15      DNS         119 Standard query response 0x6f8a  CNAME deathstar.onosproject.org A 54.241.37.8
  4 0.070210000  137.204.25.71  10.0.2.15      DNS         119 Standard query response 0x6f8a  CNAME deathstar.onosproject.org A 54.241.37.8
  5 0.071331000  10.0.2.15      54.241.37.8    TCP          74 57402→80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=204143 TSecr=0 WS=128
  6 0.250348000  54.241.37.8    10.0.2.15      TCP          60 80→57402 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
  7 0.250512000  10.0.2.15      54.241.37.8    TCP          54 57402→80 [ACK] Seq=1 Ack=1 Win=29200 Len=0
  8 0.251238000  10.0.2.15      54.241.37.8    HTTP        602 GET /1.7.0/ HTTP/1.1
  9 0.251477000  54.241.37.8    10.0.2.15      TCP          60 80→57402 [ACK] Seq=1 Ack=549 Win=65535 Len=0
 10 0.429515000  54.241.37.8    10.0.2.15      HTTP       1517 HTTP/1.1 200 OK  (text/html)
 11 0.429556000  10.0.2.15      54.241.37.8    TCP          54 57402→80 [ACK] Seq=549 Ack=1464 Win=31240 Len=0
 12 0.470601000  10.0.2.15      54.241.37.8    HTTP        666 GET /1.7.0/overview-frame.html HTTP/1.1
 13 0.470896000  54.241.37.8    10.0.2.15      TCP          60 80→57402 [ACK] Seq=1464 Ack=1161 Win=65535 Len=0
 14 0.482394000  10.0.2.15      54.241.37.8    TCP          74 57403→80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=204246 TSecr=0 WS=128
 15 0.493182000  10.0.2.15      54.241.37.8    TCP          74 57404→80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=204249 TSecr=0 WS=128
 16 0.648990000  54.241.37.8    10.0.2.15      HTTP       1934 HTTP/1.1 200 OK  (text/html)
 17 0.649012000  10.0.2.15      54.241.37.8    TCP          54 57402→80 [ACK] Seq=1161 Ack=3344 Win=35500 Len=0
```

```
▷ Frame 1: 79 bytes on wire (632 bits), 79 bytes captured (632 bits) on interface 0
▷ Ethernet II, Src: CadmusCo_ff:f7:c7 (08:00:27:ff:f7:c7), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)
▷ Internet Protocol Version 4, Src: 10.0.2.15 (10.0.2.15), Dst: 137.204.25.71 (137.204.25.71)
▷ User Datagram Protocol, Src Port: 31129 (31129), Dst Port: 53 (53)
▷ Domain Name System (query)
```

```
0000  52 54 00 12 35 02 08 00  27 ff f7 c7 08 00
0010  00 41 79 c3 40 00 40 11  11 c7 0a 00 02 0f
0020  19 47 79 99 00 35 00 2d  af 60 6f 8a 01 00
0030  00 00 00 00 00 00 03 61  70 69 0b 6f 6e 6f
0040  72 6f 6a 65 63 74 03 6f  72 67 00 00 01 00
```

File: "/Users/chiaracontoli/P... Packets: 119 · Displayed: 119 (100.0%) · Load time: 0:00.001 Profile: Default

- DNS is an application level protocol
- DNS is at the same level as HTTP
- Instead of using TCP as transport it uses UDP

To get more information

- Sending request to a DNS server
  - *dig*
  - *nslookup* (interactive mode and non)
- Several type of requests can be performed
  - Given a name, find the IP address
  - Find a mail server linked to a name
- Most common Queries:
  - *A* (default query): given the host name, it returns the IP address
  - *ANY*: returns all DNS field associated to the address
  - *PTR*: given the IP address, returns the host name
  - *MX*: return the mail server associated to the domain name
- Example: nslookup -querytype=ANY www.cisco.com

To get more information

- `dig` : analyze a DNS request

```
$ dig www.cs.ucla.edu

; <<>> DiG 9.8.3-P1 <<>> www.cs.ucla.edu
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 5002
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.cs.ucla.edu.               IN      A

;; ANSWER SECTION:
www.cs.ucla.edu.      12243     IN      A       164.67.100.181

;; Query time: 66 msec
;; SERVER: 192.168.43.1#53(192.168.43.1)
;; WHEN: Sun Mar 19 20:03:55 2017
;; MSG SIZE  rcvd: 49
```

To get more information

- Examples
  - nslookup -querytype=ANY www.cs.unibo.it
  - dig @ns1.garr.net cs.unibo.it ANY