Lab 2

CS-335a

Fall 2012
Computer Science Department

Manolis Surligas
surligas@csd.uoc.gr

Summary

- At this lab we will cover:
  - Basics of Transport Layer (TCP, UDP)
  - Broadcast
  - ARP
  - DNS
  - More Wireshark filters
  - Several very useful network tools

Transport Layer

- Two major protocols
  - TCP providing reliable communication
  - UDP providing fast but unreliable communication

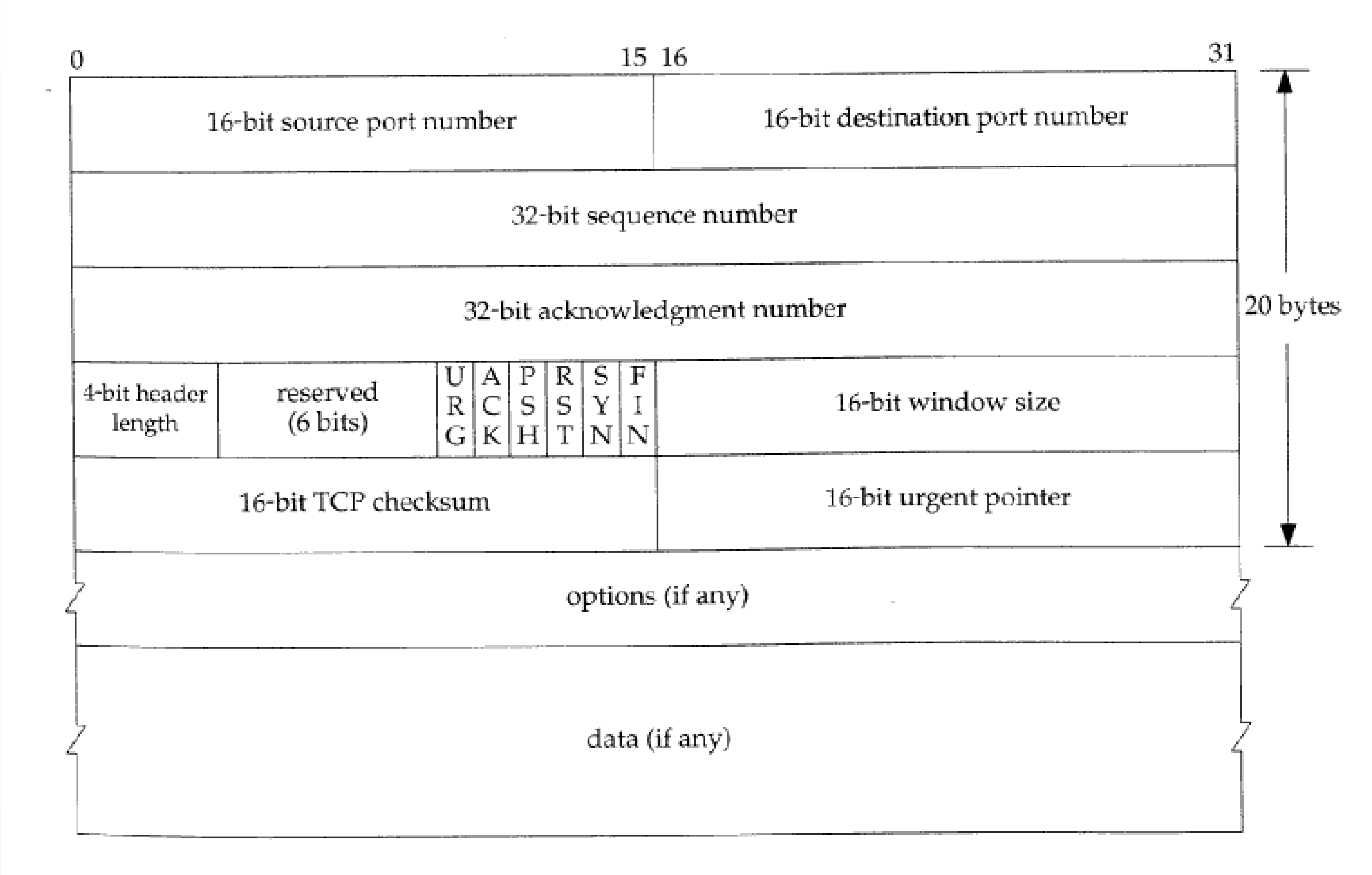
- Other protocols:
  - DCCP
  - SCTP
  - RSVP

Transport Layer

- Link layer has as source and destination identifier, the MAC addresses of the source and destination node
- Network layer respectively, uses the IP address
- Transport layer uses port numbers
- Two port numbers in every packet
  - Source port
  - Destination port

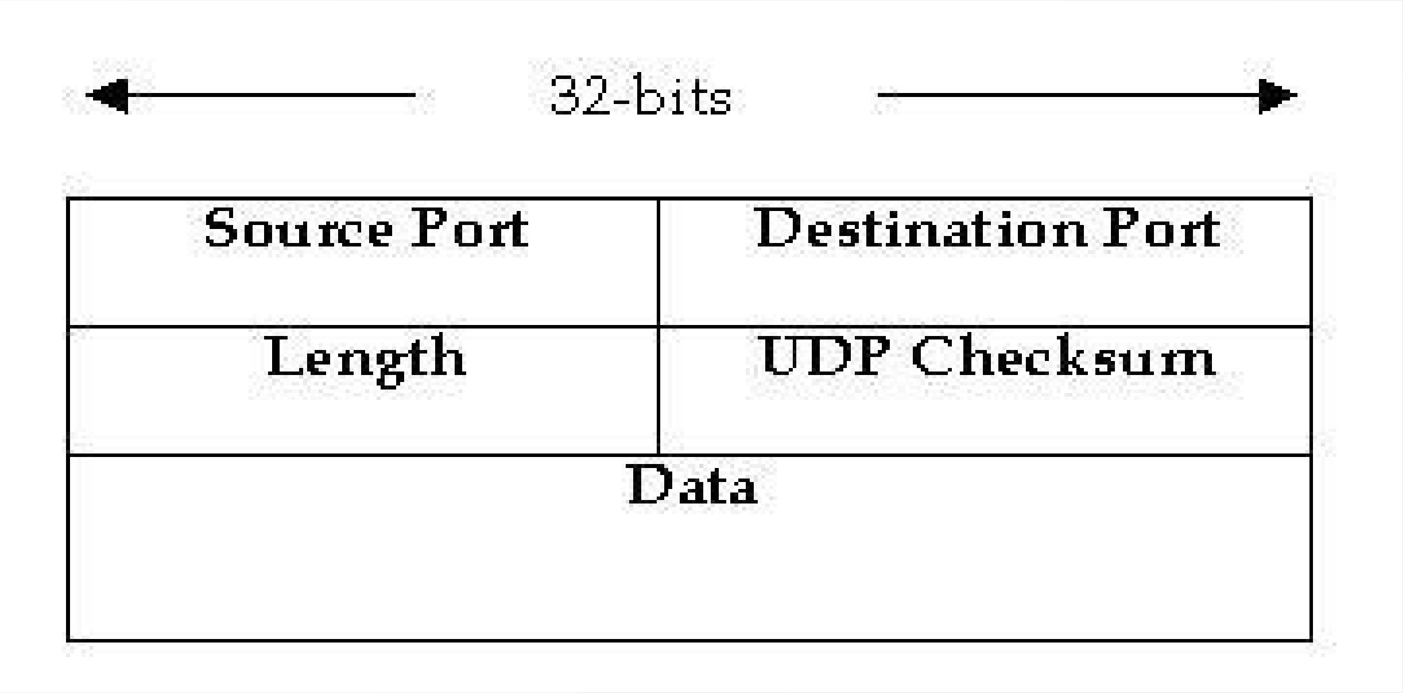
Transport Layer: TCP

- TCP provides reliable communication avoiding data corruption due to packet losses



Transport Layer: UDP

- UDP is a simple protocol providing unreliable but fast data transfer
- Used in realtime applications
  - VoIP
  - Online games
  - Streaming



32-bits

Source Port	Destination Port
Length	UDP Checksum
Data	

Transport Layer

- From TCP and UDP headers we see that each port is 16-bits long
- 2^16 different ports
- Again not all of them are available
- IANA keeps track for the reserved ports and the services that use them
  http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xml
- Some well known ports:
  - 22: ssh
  - 80, 8080: http
  - 443 https

Broadcast

- The last IP of a subnet is used for broadcasting
- A broadcast message is received by all hosts in the same subnet
- Only UDP can be used in broadcast messages
- Eg:
  - Broadcast message to 192.168.1.255, will be received by all hosts in the 192.168.1.0/24 network
  - Another broadcast to 192.168.255.255, will be received by all hosts in the 192.168.0.0/16 network

Broadcast

- Theoretically a broadcast to the 255.255.255.255, will be received by all hosts in the internet
- At most cases, routers do not allow broadcasting outside our subnet, in order to prevent flooding
- Broadcast is used:
  - By many protocols like ARP, DHCP, etc
  - For automatic server discovery
  - For easily delivering messages to all hosts in the subnet
- IPv6 does not implement broadcast. It uses multicast instead

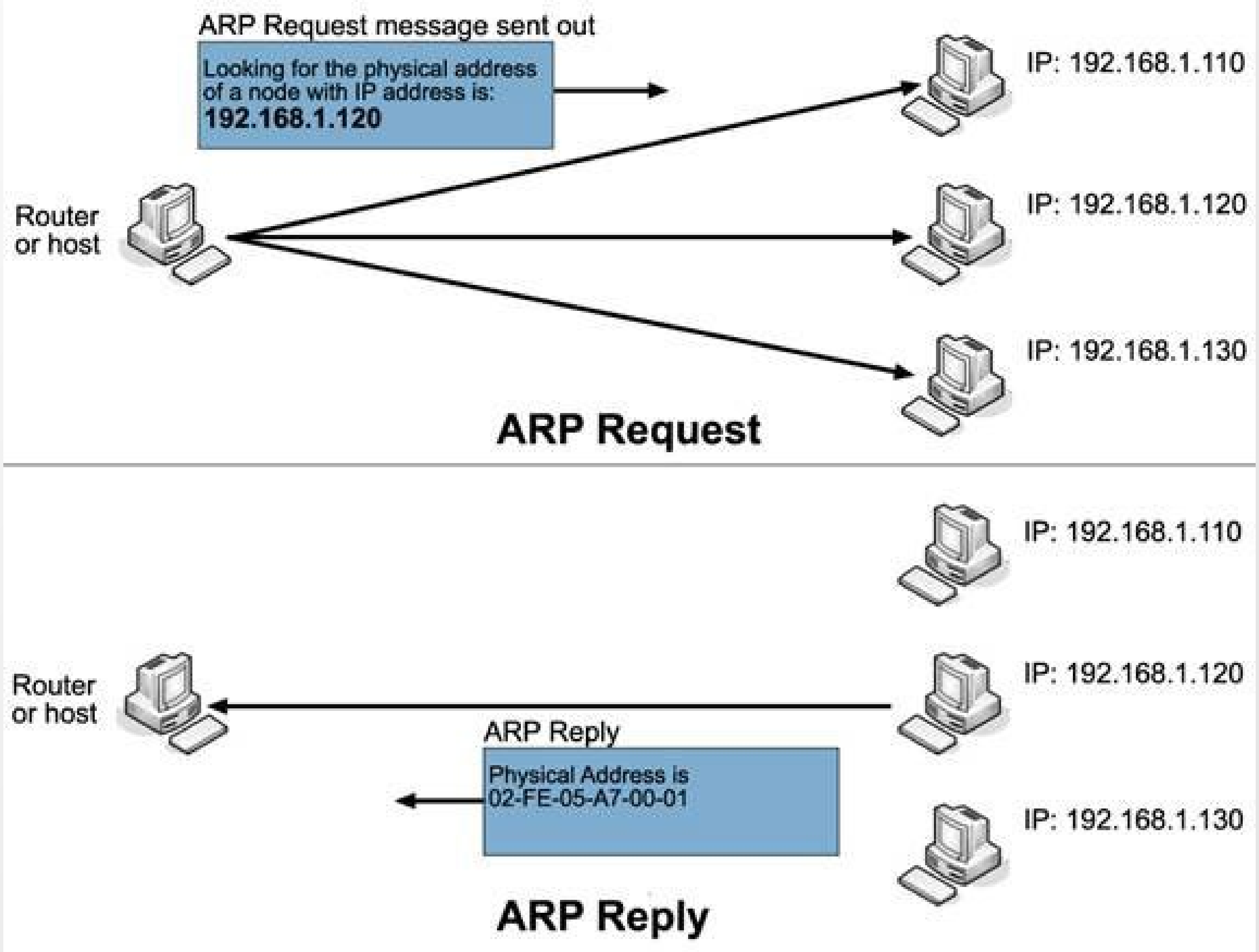
Address Resolution Protocol (ARP)

- Host A wants to sent a packet to B. A knows the IP of B. But A needs also the MAC of B! (Or the MAC of the next node if A and B are not direct connected)

- ARP helps to retrieve the MAC of a host with a known IP
  - It checks the arp cache table if the MAC of a given IP exists
  - If yes, it uses it immediately
  - If know, host A broadcasts a message says “Who has the IP x”
  - The node with the IP x replies back his MAC

- ARP cache table can be viewed with the *arp* command. Its values are periodically cleared

Address Resolution Protocol (ARP)



ARP Request message sent out

Looking for the physical address of a node with IP address is:
192.168.1.120

ARP Request

ARP Reply

Physical Address is
02-FE-05-A7-00-01

ARP Reply

Domain Name System (DNS)

- Translate human friendly host names into IP addresses
- It is easier to remember www.csd.uoc.gr rather than 147.52.78.3
- The server of CSD can change (due to a fault, or for maintenance) to another backup machine, without changing the URL of the website
- Works like the phone book, but for URLs
- Many DNS servers across the internet, structured in hierarchical form

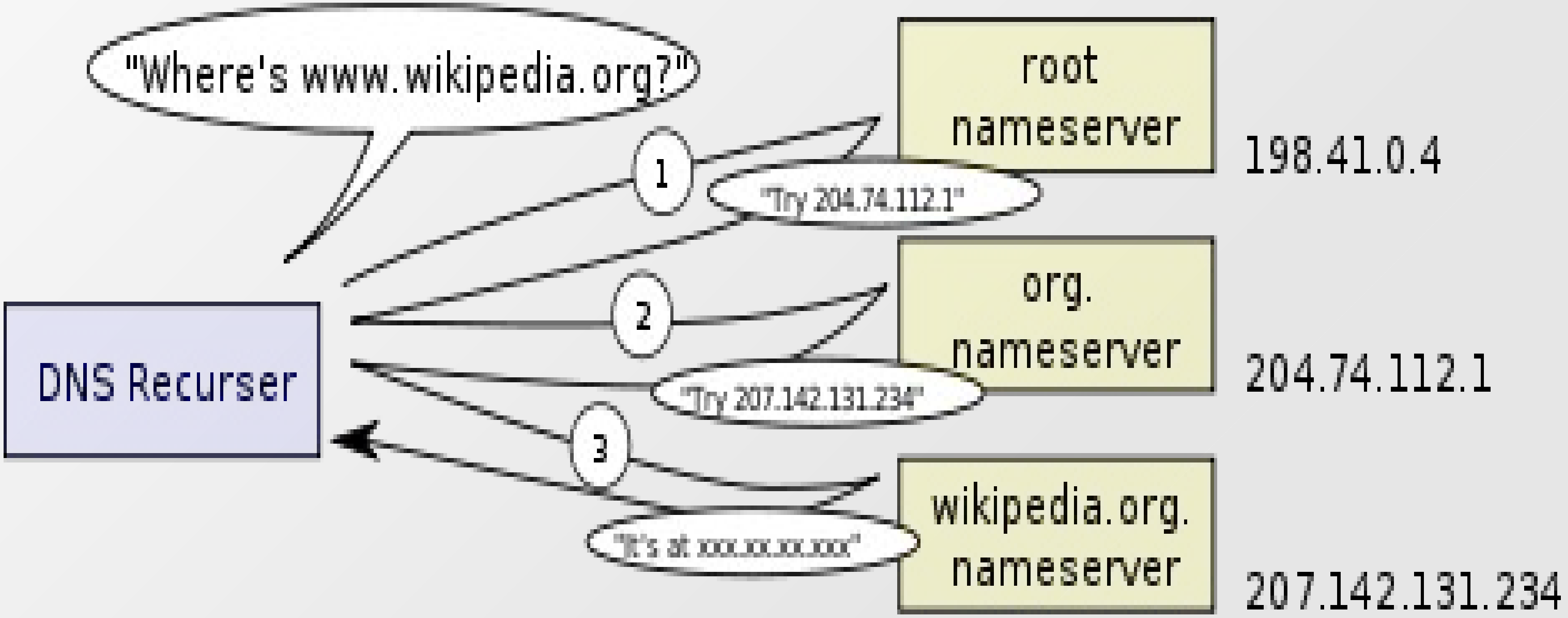
Domain Name System (DNS)

- The file containing your DNS server is the */etc/resolv.conf*
  - *Contains one or more DNS servers in priority order, or the IP of the gateway that is responsible to provide a DNS server*

- Host A wants to get the IP of wikipedia
  - Host A queries his DNS server
  - If the DNS server do not have an entry for wikipedia, returns a server that may have it
  - The procedure is repeated until a DNS server return the IP of the website

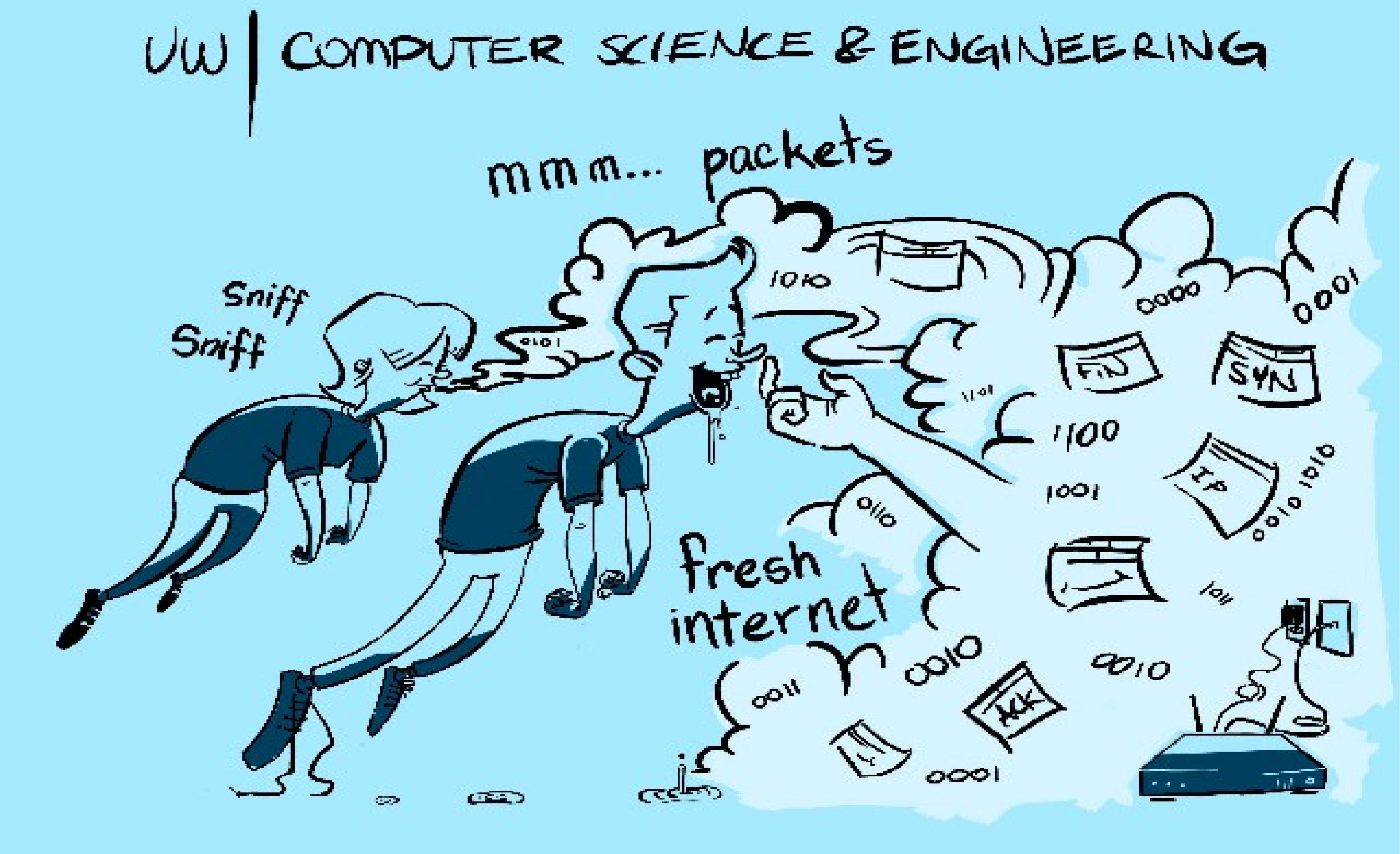
- We will see more about the DNS system, later, using the *dig* tool

Domain Name System (DNS)

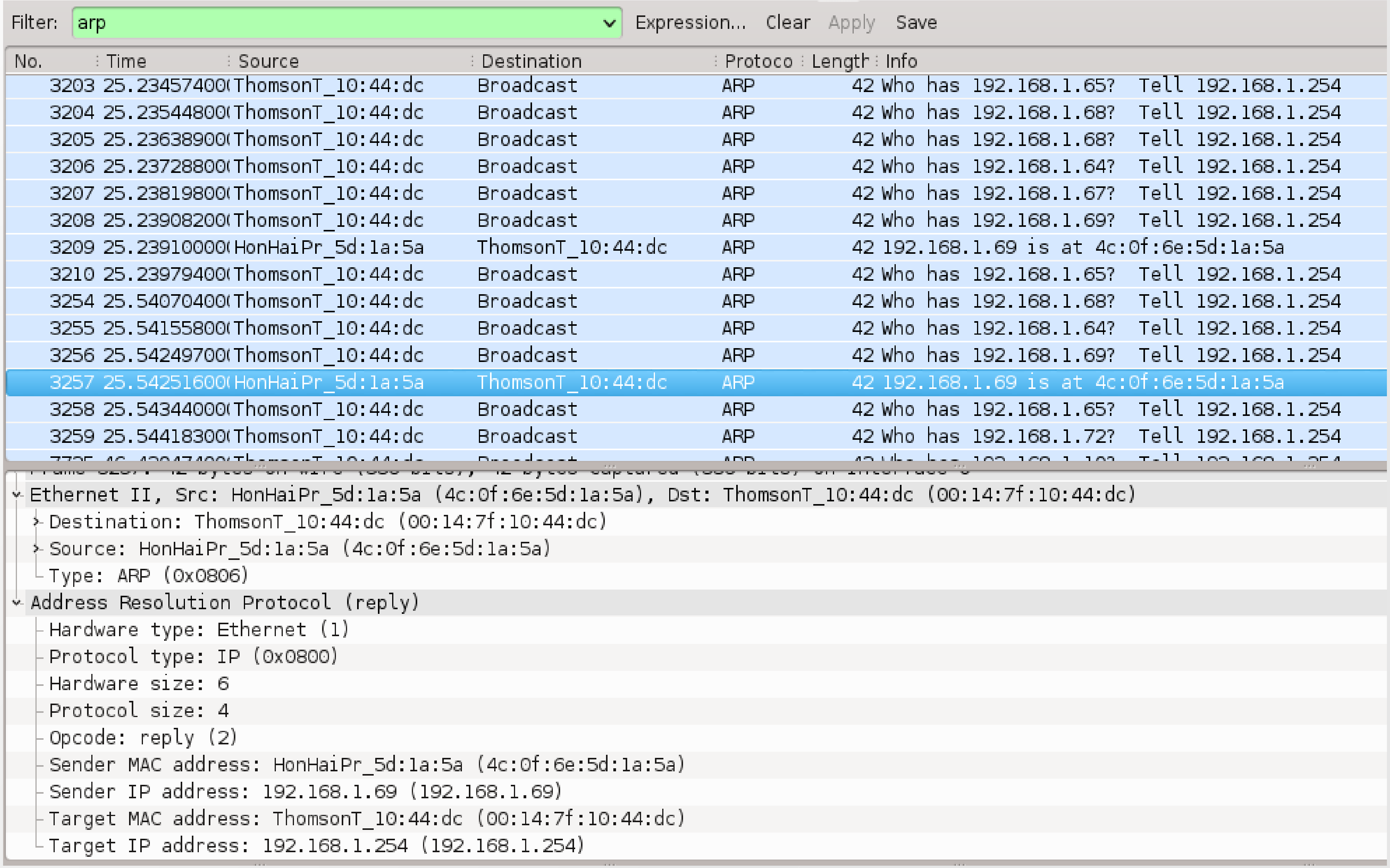


More Wireshark!

- Now that we know some protocols and we are familiar with Transport layer lets perform some interesting filters!!!



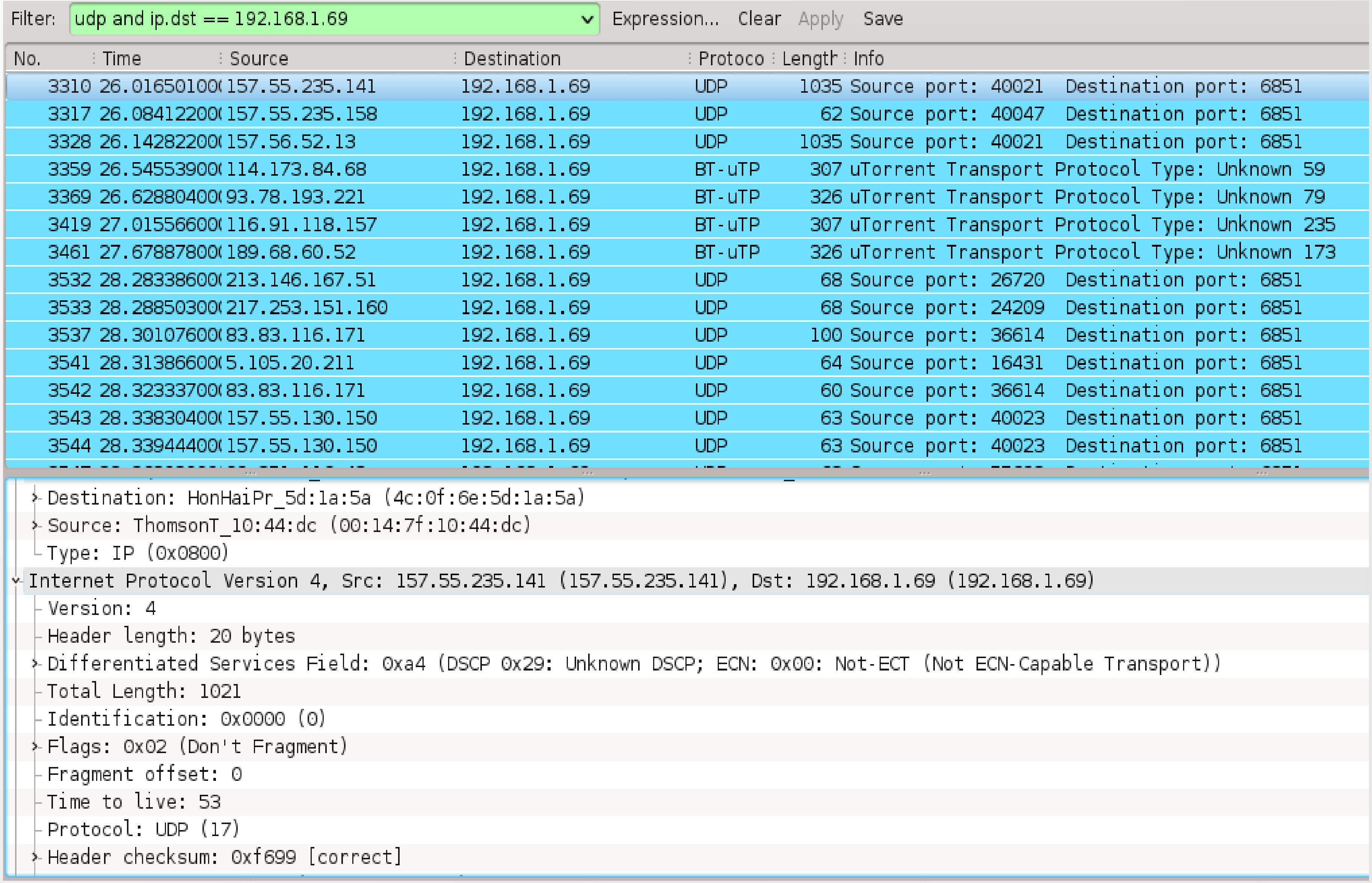
ARP packets



No.	Time	Source	Destination	Protoco	Length	Info
3203	25.23457400(	ThomsonT_10:44:dc	Broadcast	ARP	42	Who has 192.168.1.65? Tell 192.168.1.254
3204	25.23544800(	ThomsonT_10:44:dc	Broadcast	ARP	42	Who has 192.168.1.68? Tell 192.168.1.254
3205	25.23638900(	ThomsonT_10:44:dc	Broadcast	ARP	42	Who has 192.168.1.68? Tell 192.168.1.254
3206	25.23728800(	ThomsonT_10:44:dc	Broadcast	ARP	42	Who has 192.168.1.64? Tell 192.168.1.254
3207	25.23819800(	ThomsonT_10:44:dc	Broadcast	ARP	42	Who has 192.168.1.67? Tell 192.168.1.254
3208	25.23908200(	ThomsonT_10:44:dc	Broadcast	ARP	42	Who has 192.168.1.69? Tell 192.168.1.254
3209	25.23910000(	HonHaiPr_5d:1a:5a	ThomsonT_10:44:dc	ARP	42	192.168.1.69 is at 4c:0f:6e:5d:1a:5a
3210	25.23979400(	ThomsonT_10:44:dc	Broadcast	ARP	42	Who has 192.168.1.65? Tell 192.168.1.254
3254	25.54070400(	ThomsonT_10:44:dc	Broadcast	ARP	42	Who has 192.168.1.68? Tell 192.168.1.254
3255	25.54155800(	ThomsonT_10:44:dc	Broadcast	ARP	42	Who has 192.168.1.64? Tell 192.168.1.254
3256	25.54249700(	ThomsonT_10:44:dc	Broadcast	ARP	42	Who has 192.168.1.69? Tell 192.168.1.254
3257	25.54251600(	HonHaiPr_5d:1a:5a	ThomsonT_10:44:dc	ARP	42	192.168.1.69 is at 4c:0f:6e:5d:1a:5a
3258	25.54344000(	ThomsonT_10:44:dc	Broadcast	ARP	42	Who has 192.168.1.65? Tell 192.168.1.254
3259	25.54418300(	ThomsonT_10:44:dc	Broadcast	ARP	42	Who has 192.168.1.72? Tell 192.168.1.254

```
Ethernet II, Src: HonHaiPr_5d:1a:5a (4c:0f:6e:5d:1a:5a), Dst: ThomsonT_10:44:dc (00:14:7f:10:44:dc)
  Destination: ThomsonT_10:44:dc (00:14:7f:10:44:dc)
  Source: HonHaiPr_5d:1a:5a (4c:0f:6e:5d:1a:5a)
  Type: ARP (0x0806)
Address Resolution Protocol (reply)
  Hardware type: Ethernet (1)
  Protocol type: IP (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: reply (2)
  Sender MAC address: HonHaiPr_5d:1a:5a (4c:0f:6e:5d:1a:5a)
  Sender IP address: 192.168.1.69 (192.168.1.69)
  Target MAC address: ThomsonT_10:44:dc (00:14:7f:10:44:dc)
  Target IP address: 192.168.1.254 (192.168.1.254)
```

UDP with specific destination IP

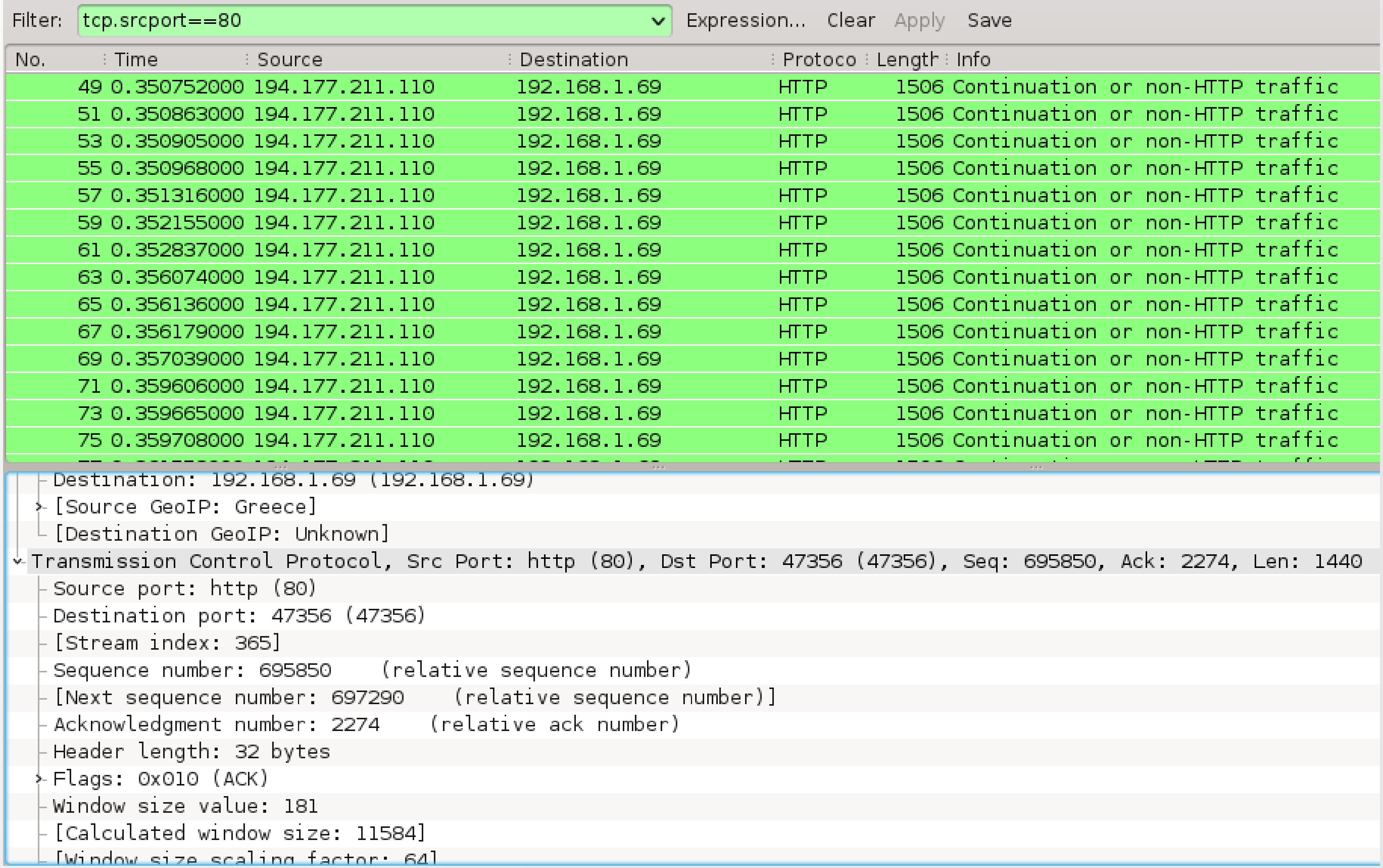


Filter: udp and ip.dst == 192.168.1.69

No.	Time	Source	Destination	Protoco	Lengtr	Info
3310	26.01650100(	157.55.235.141	192.168.1.69	UDP	1035	Source port: 40021 Destination port: 6851
3317	26.08412200(	157.55.235.158	192.168.1.69	UDP	62	Source port: 40047 Destination port: 6851
3328	26.14282200(	157.56.52.13	192.168.1.69	UDP	1035	Source port: 40021 Destination port: 6851
3359	26.54553900(	114.173.84.68	192.168.1.69	BT-uTP	307	uTorrent Transport Protocol Type: Unknown 59
3369	26.62880400(	93.78.193.221	192.168.1.69	BT-uTP	326	uTorrent Transport Protocol Type: Unknown 79
3419	27.01556600(	116.91.118.157	192.168.1.69	BT-uTP	307	uTorrent Transport Protocol Type: Unknown 235
3461	27.67887800(	189.68.60.52	192.168.1.69	BT-uTP	326	uTorrent Transport Protocol Type: Unknown 173
3532	28.28338600(	213.146.167.51	192.168.1.69	UDP	68	Source port: 26720 Destination port: 6851
3533	28.28850300(	217.253.151.160	192.168.1.69	UDP	68	Source port: 24209 Destination port: 6851
3537	28.30107600(	83.83.116.171	192.168.1.69	UDP	100	Source port: 36614 Destination port: 6851
3541	28.31386600(	5.105.20.211	192.168.1.69	UDP	64	Source port: 16431 Destination port: 6851
3542	28.32333700(	83.83.116.171	192.168.1.69	UDP	60	Source port: 36614 Destination port: 6851
3543	28.33830400(	157.55.130.150	192.168.1.69	UDP	63	Source port: 40023 Destination port: 6851
3544	28.33944400(	157.55.130.150	192.168.1.69	UDP	63	Source port: 40023 Destination port: 6851

```
 > Destination: HonHaiPr_5d:1a:5a (4c:0f:6e:5d:1a:5a)
 > Source: ThomsonT_10:44:dc (00:14:7f:10:44:dc)
   Type: IP (0x0800)
v Internet Protocol Version 4, Src: 157.55.235.141 (157.55.235.141), Dst: 192.168.1.69 (192.168.1.69)
   Version: 4
   Header length: 20 bytes
 > Differentiated Services Field: 0xa4 (DSCP 0x29: Unknown DSCP; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
   Total Length: 1021
   Identification: 0x0000 (0)
 > Flags: 0x02 (Don't Fragment)
   Fragment offset: 0
   Time to live: 53
   Protocol: UDP (17)
 > Header checksum: 0xf699 [correct]
```

TCP with specific source port

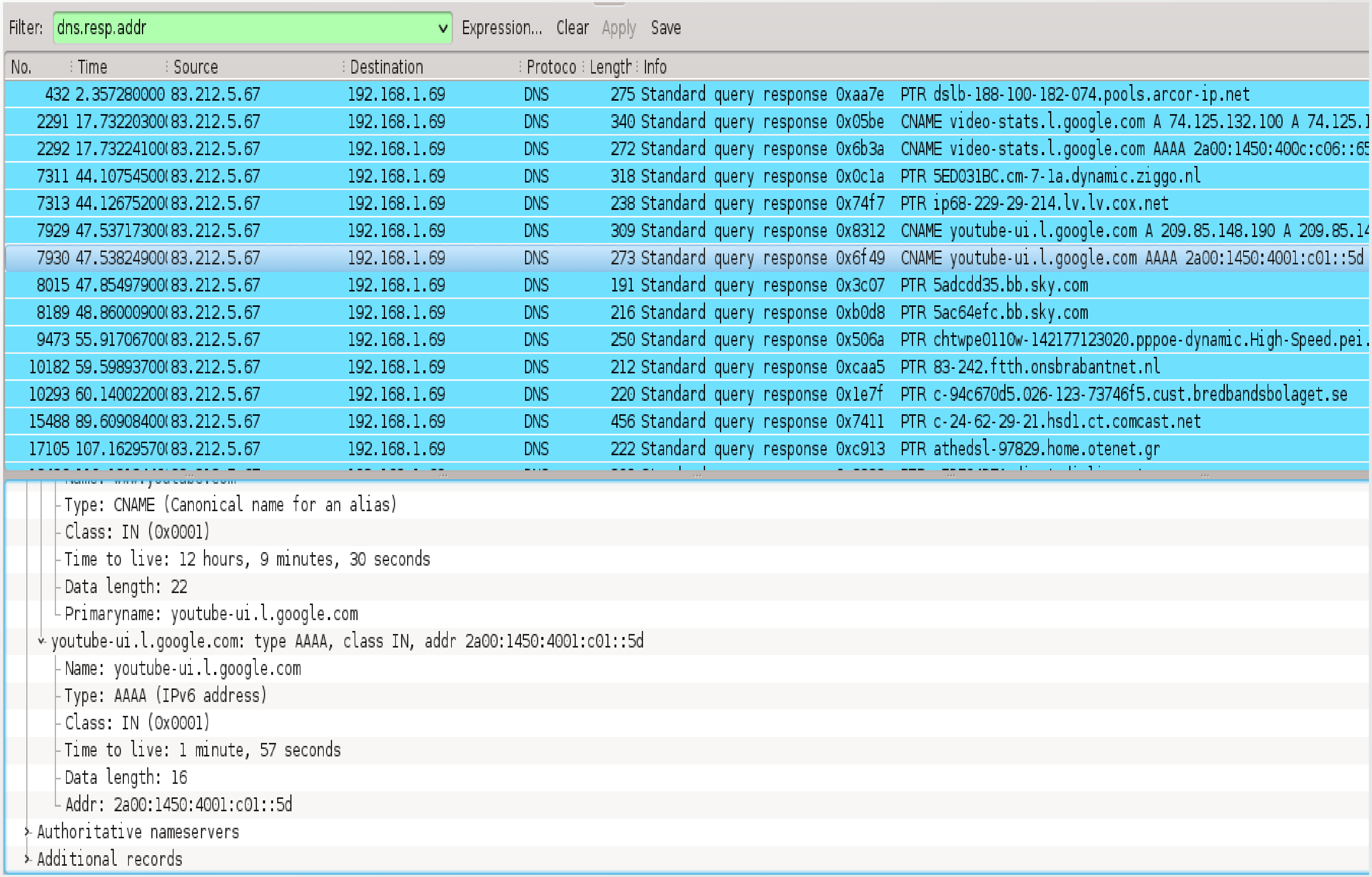


Filter: tcp.srcport==80

```
No.      Time        Source            Destination      Protoco Length Info
      49 0.350752000 194.177.211.110   192.168.1.69     HTTP      1506 Continuation or non-HTTP traffic
      51 0.350863000 194.177.211.110   192.168.1.69     HTTP      1506 Continuation or non-HTTP traffic
      53 0.350905000 194.177.211.110   192.168.1.69     HTTP      1506 Continuation or non-HTTP traffic
      55 0.350968000 194.177.211.110   192.168.1.69     HTTP      1506 Continuation or non-HTTP traffic
      57 0.351316000 194.177.211.110   192.168.1.69     HTTP      1506 Continuation or non-HTTP traffic
      59 0.352155000 194.177.211.110   192.168.1.69     HTTP      1506 Continuation or non-HTTP traffic
      61 0.352837000 194.177.211.110   192.168.1.69     HTTP      1506 Continuation or non-HTTP traffic
      63 0.356074000 194.177.211.110   192.168.1.69     HTTP      1506 Continuation or non-HTTP traffic
      65 0.356136000 194.177.211.110   192.168.1.69     HTTP      1506 Continuation or non-HTTP traffic
      67 0.356179000 194.177.211.110   192.168.1.69     HTTP      1506 Continuation or non-HTTP traffic
      69 0.357039000 194.177.211.110   192.168.1.69     HTTP      1506 Continuation or non-HTTP traffic
      71 0.359606000 194.177.211.110   192.168.1.69     HTTP      1506 Continuation or non-HTTP traffic
      73 0.359665000 194.177.211.110   192.168.1.69     HTTP      1506 Continuation or non-HTTP traffic
      75 0.359708000 194.177.211.110   192.168.1.69     HTTP      1506 Continuation or non-HTTP traffic
```

```
   ├ Destination: 192.168.1.69 (192.168.1.69)
   ├ [Source GeoIP: Greece]
   └ [Destination GeoIP: Unknown]
 ˅ Transmission Control Protocol, Src Port: http (80), Dst Port: 47356 (47356), Seq: 695850, Ack: 2274, Len: 1440
   ├ Source port: http (80)
   ├ Destination port: 47356 (47356)
   ├ [Stream index: 365]
   ├ Sequence number: 695850    (relative sequence number)
   ├ [Next sequence number: 697290    (relative sequence number)]
   ├ Acknowledgment number: 2274    (relative ack number)
   ├ Header length: 32 bytes
   ├ Flags: 0x010 (ACK)
   ├ Window size value: 181
   ├ [Calculated window size: 11584]
   ├ [Window size scaling factor: 64]
```

DNS responses



```
No.     Time        Source        Destination    Protoco Length Info
  432 2.357280000 83.212.5.67    192.168.1.69   DNS    275 Standard query response 0xaa7e  PTR dslb-188-100-182-074.pools.arcor-ip.net
 2291 17.73220300 83.212.5.67    192.168.1.69   DNS    340 Standard query response 0x05be  CNAME video-stats.l.google.com A 74.125.132.100 A 74.125.1
 2292 17.73224100 83.212.5.67    192.168.1.69   DNS    272 Standard query response 0x6b3a  CNAME video-stats.l.google.com AAAA 2a00:1450:400c:c06::65
 7311 44.10754500 83.212.5.67    192.168.1.69   DNS    318 Standard query response 0x0c1a  PTR 5ED031BC.cm-7-1a.dynamic.ziggo.nl
 7313 44.12675200 83.212.5.67    192.168.1.69   DNS    238 Standard query response 0x74f7  PTR ip68-229-29-214.lv.lv.cox.net
 7929 47.53717300 83.212.5.67    192.168.1.69   DNS    309 Standard query response 0x8312  CNAME youtube-ui.l.google.com A 209.85.148.190 A 209.85.14
 7930 47.53824900 83.212.5.67    192.168.1.69   DNS    273 Standard query response 0x6f49  CNAME youtube-ui.l.google.com AAAA 2a00:1450:4001:c01::5d
 8015 47.85497900 83.212.5.67    192.168.1.69   DNS    191 Standard query response 0x3c07  PTR 5adcdd35.bb.sky.com
 8189 48.86000900 83.212.5.67    192.168.1.69   DNS    216 Standard query response 0xb0d8  PTR 5ac64efc.bb.sky.com
 9473 55.91706700 83.212.5.67    192.168.1.69   DNS    250 Standard query response 0x506a  PTR chtwpe0110w-142177123020.pppoe-dynamic.High-Speed.pei.
10182 59.59893700 83.212.5.67    192.168.1.69   DNS    212 Standard query response 0xcaa5  PTR 83-242.ftth.onsbrabantnet.nl
10293 60.14002200 83.212.5.67    192.168.1.69   DNS    220 Standard query response 0x1e7f  PTR c-94c670d5.026-123-73746f5.cust.bredbandsbolaget.se
15488 89.60908400 83.212.5.67    192.168.1.69   DNS    456 Standard query response 0x7411  PTR c-24-62-29-21.hsd1.ct.comcast.net
17105 107.1629570 83.212.5.67    192.168.1.69   DNS    222 Standard query response 0xc913  PTR athedsl-97829.home.otenet.gr
```

```
        Name: www.youtube.com
        Type: CNAME (Canonical name for an alias)
        Class: IN (0x0001)
        Time to live: 12 hours, 9 minutes, 30 seconds
        Data length: 22
        Primaryname: youtube-ui.l.google.com
     youtube-ui.l.google.com: type AAAA, class IN, addr 2a00:1450:4001:c01::5d
        Name: youtube-ui.l.google.com
        Type: AAAA (IPv6 address)
        Class: IN (0x0001)
        Time to live: 1 minute, 57 seconds
        Data length: 16
        Addr: 2a00:1450:4001:c01::5d
  Authoritative nameservers
  Additional records
```

Useful Network tools

- For debugging, testing, experimenting etc, there are many useful network tools
  - netstat
  - iftop
  - ping
  - traceroute
  - dig
  - GeoIP

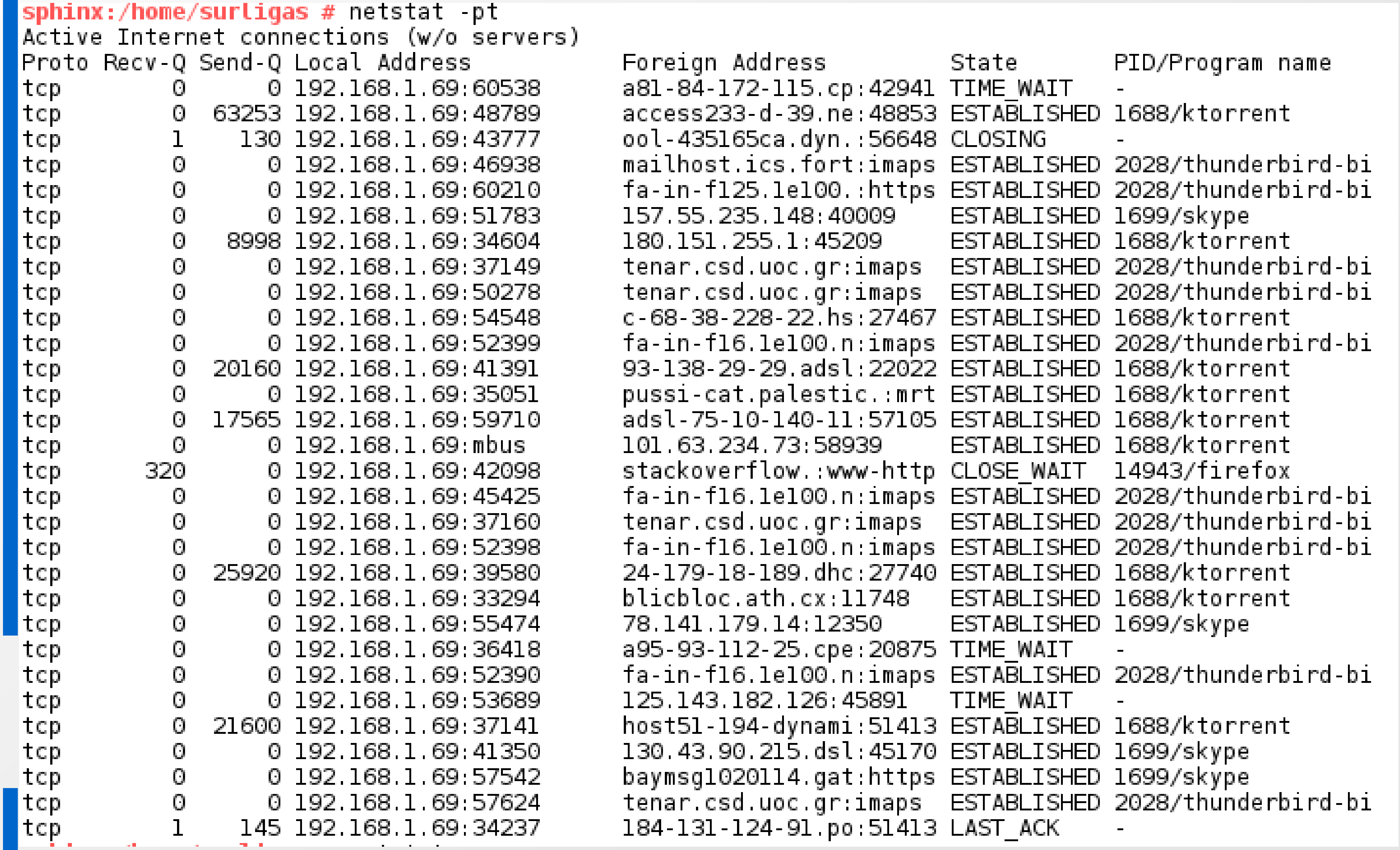
Netstat

- Netstat can show several info about the open network connections:
  - Which programs have network connections
  - What transfer protocol they are use
  - What ports they use
  - Statistics per transfer protocol

- It is a very useful tool when we are writing our own programs

- Frequently used parameters
  - netstat -pt: Show programs that have open TCP connections
  - netstat -pu: Same with the above, for UDP
- For more info: *man netstat*

Netstat



```
sphinx:/home/surligas # netstat -pt
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 192.168.1.69:60538      a81-84-172-115.cp:42941 TIME_WAIT   -
tcp        0  63253 192.168.1.69:48789      access233-d-39.ne:48853 ESTABLISHED 1688/ktorrent
tcp        1    130 192.168.1.69:43777      ool-435165ca.dyn.:56648 CLOSING     -
tcp        0      0 192.168.1.69:46938      mailhost.ics.fort:imaps ESTABLISHED 2028/thunderbird-bi
tcp        0      0 192.168.1.69:60210      fa-in-f125.1e100.:https ESTABLISHED 2028/thunderbird-bi
tcp        0      0 192.168.1.69:51783      157.55.235.148:40009    ESTABLISHED 1699/skype
tcp        0   8998 192.168.1.69:34604      180.151.255.1:45209     ESTABLISHED 1688/ktorrent
tcp        0      0 192.168.1.69:37149      tenar.csd.uoc.gr:imaps  ESTABLISHED 2028/thunderbird-bi
tcp        0      0 192.168.1.69:50278      tenar.csd.uoc.gr:imaps  ESTABLISHED 2028/thunderbird-bi
tcp        0      0 192.168.1.69:54548      c-68-38-228-22.hs:27467 ESTABLISHED 1688/ktorrent
tcp        0      0 192.168.1.69:52399      fa-in-f16.1e100.n:imaps ESTABLISHED 2028/thunderbird-bi
tcp        0  20160 192.168.1.69:41391      93-138-29-29.adsl:22022 ESTABLISHED 1688/ktorrent
tcp        0      0 192.168.1.69:35051      pussi-cat.palestic.:mrt ESTABLISHED 1688/ktorrent
tcp        0  17565 192.168.1.69:59710      adsl-75-10-140-11:57105 ESTABLISHED 1688/ktorrent
tcp        0      0 192.168.1.69:mbus       101.63.234.73:58939     ESTABLISHED 1688/ktorrent
tcp      320      0 192.168.1.69:42098      stackoverflow.:www-http CLOSE_WAIT  14943/firefox
tcp        0      0 192.168.1.69:45425      fa-in-f16.1e100.n:imaps ESTABLISHED 2028/thunderbird-bi
tcp        0      0 192.168.1.69:37160      tenar.csd.uoc.gr:imaps  ESTABLISHED 2028/thunderbird-bi
tcp        0      0 192.168.1.69:52398      fa-in-f16.1e100.n:imaps ESTABLISHED 2028/thunderbird-bi
tcp        0  25920 192.168.1.69:39580      24-179-18-189.dhc:27740 ESTABLISHED 1688/ktorrent
tcp        0      0 192.168.1.69:33294      blicbloc.ath.cx:11748   ESTABLISHED 1688/ktorrent
tcp        0      0 192.168.1.69:55474      78.141.179.14:12350     ESTABLISHED 1699/skype
tcp        0      0 192.168.1.69:36418      a95-93-112-25.cpe:20875 TIME_WAIT   -
tcp        0      0 192.168.1.69:52390      fa-in-f16.1e100.n:imaps ESTABLISHED 2028/thunderbird-bi
tcp        0      0 192.168.1.69:53689      125.143.182.126:45891   TIME_WAIT   -
tcp        0  21600 192.168.1.69:37141      host51-194-dynami:51413 ESTABLISHED 1688/ktorrent
tcp        0      0 192.168.1.69:41350      130.43.90.215.dsl:45170 ESTABLISHED 1699/skype
tcp        0      0 192.168.1.69:57542      baymsg1020114.gat:https ESTABLISHED 1699/skype
tcp        0      0 192.168.1.69:57624      tenar.csd.uoc.gr:imaps  ESTABLISHED 2028/thunderbird-bi
tcp        1    145 192.168.1.69:34237      184-131-124-91.po:51413 LAST_ACK    -
```

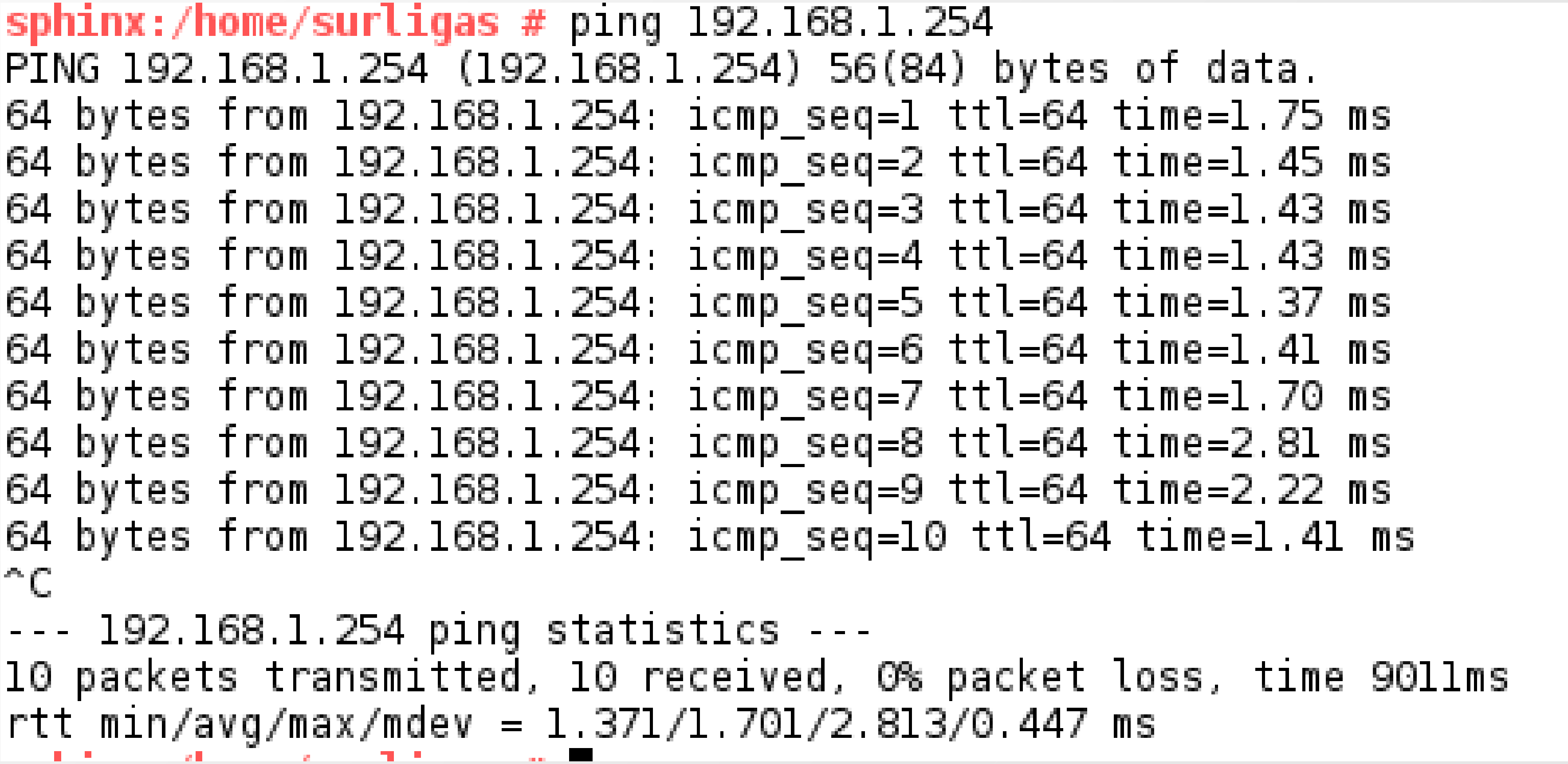
iftop

- iftop is a console tool that shows realtime information about the bandwith that is spent for each network connection
- Useful to identify which is the current incoming/outgoing data rate, or
- In combination with netstat, someone can identify which program has the highest/lowest data rate

Ping

- One of the first network tools
- Used today to identify if a host is up or not
  - But this not always work. There are hosts that do not respond to ping requests
- Used also to get the RTT (round trip time) from a specific host
- Uses the ICMP protocol
- Easy to use: *ping ip_address*
- For more info: *man ping*

Ping



```
sphinx:/home/surligas # ping 192.168.1.254
PING 192.168.1.254 (192.168.1.254) 56(84) bytes of data.
64 bytes from 192.168.1.254: icmp_seq=1 ttl=64 time=1.75 ms
64 bytes from 192.168.1.254: icmp_seq=2 ttl=64 time=1.45 ms
64 bytes from 192.168.1.254: icmp_seq=3 ttl=64 time=1.43 ms
64 bytes from 192.168.1.254: icmp_seq=4 ttl=64 time=1.43 ms
64 bytes from 192.168.1.254: icmp_seq=5 ttl=64 time=1.37 ms
64 bytes from 192.168.1.254: icmp_seq=6 ttl=64 time=1.41 ms
64 bytes from 192.168.1.254: icmp_seq=7 ttl=64 time=1.70 ms
64 bytes from 192.168.1.254: icmp_seq=8 ttl=64 time=2.81 ms
64 bytes from 192.168.1.254: icmp_seq=9 ttl=64 time=2.22 ms
64 bytes from 192.168.1.254: icmp_seq=10 ttl=64 time=1.41 ms
^C
--- 192.168.1.254 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9011ms
rtt min/avg/max/mdev = 1.371/1.701/2.813/0.447 ms
```

Traceroute

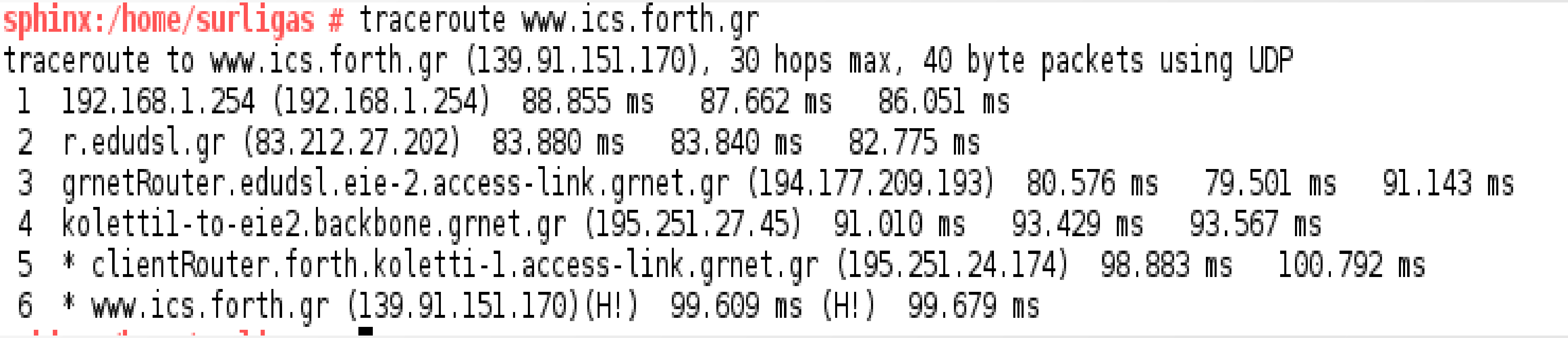
- Another very usefull tool that shows the path that packets follow until they reach the destination

- Like ping, it uses also ICMP packets (can be changed with options)

- How it works:
  - Sends first packet with TTL 1
  - Send the next packet with TTL 2,3,etc
  - A node that sees a TTL 0, reports back its IP address
  - The procedure is repeated until the final destination is reached
- With the above simple way we can find all(?) the intermediate nodes

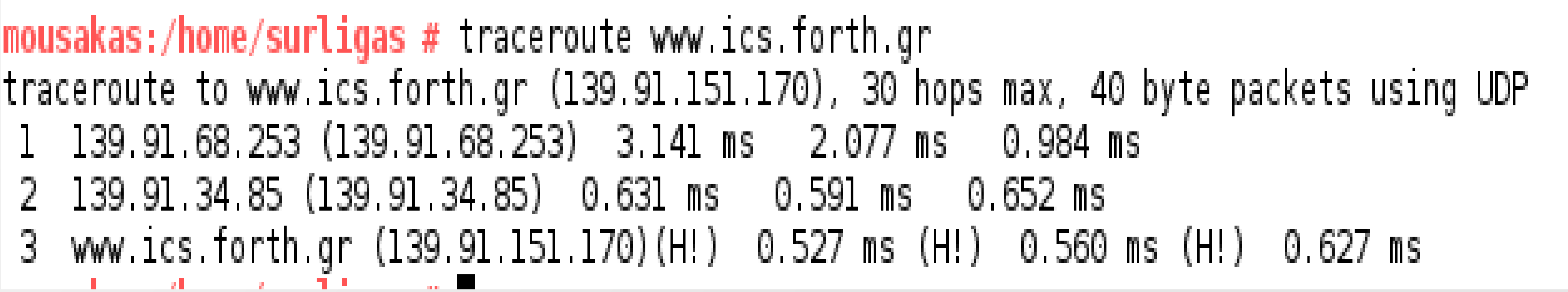
Traceroute

- Host A to www.ics.forth.gr



```
sphinx:/home/surligas # traceroute www.ics.forth.gr
traceroute to www.ics.forth.gr (139.91.151.170), 30 hops max, 40 byte packets using UDP
 1  192.168.1.254 (192.168.1.254)  88.855 ms   87.662 ms   86.051 ms
 2  r.edudsl.gr (83.212.27.202)  83.880 ms   83.840 ms   82.775 ms
 3  grnetRouter.edudsl.eie-2.access-link.grnet.gr (194.177.209.193)  80.576 ms   79.501 ms   91.143 ms
 4  kolettil-to-eie2.backbone.grnet.gr (195.251.27.45)  91.010 ms   93.429 ms   93.567 ms
 5  * clientRouter.forth.koletti-1.access-link.grnet.gr (195.251.24.174)  98.883 ms   100.792 ms
 6  * www.ics.forth.gr (139.91.151.170)(H!)  99.609 ms (H!)  99.679 ms
```

- Host B to www.ics.forth.gr

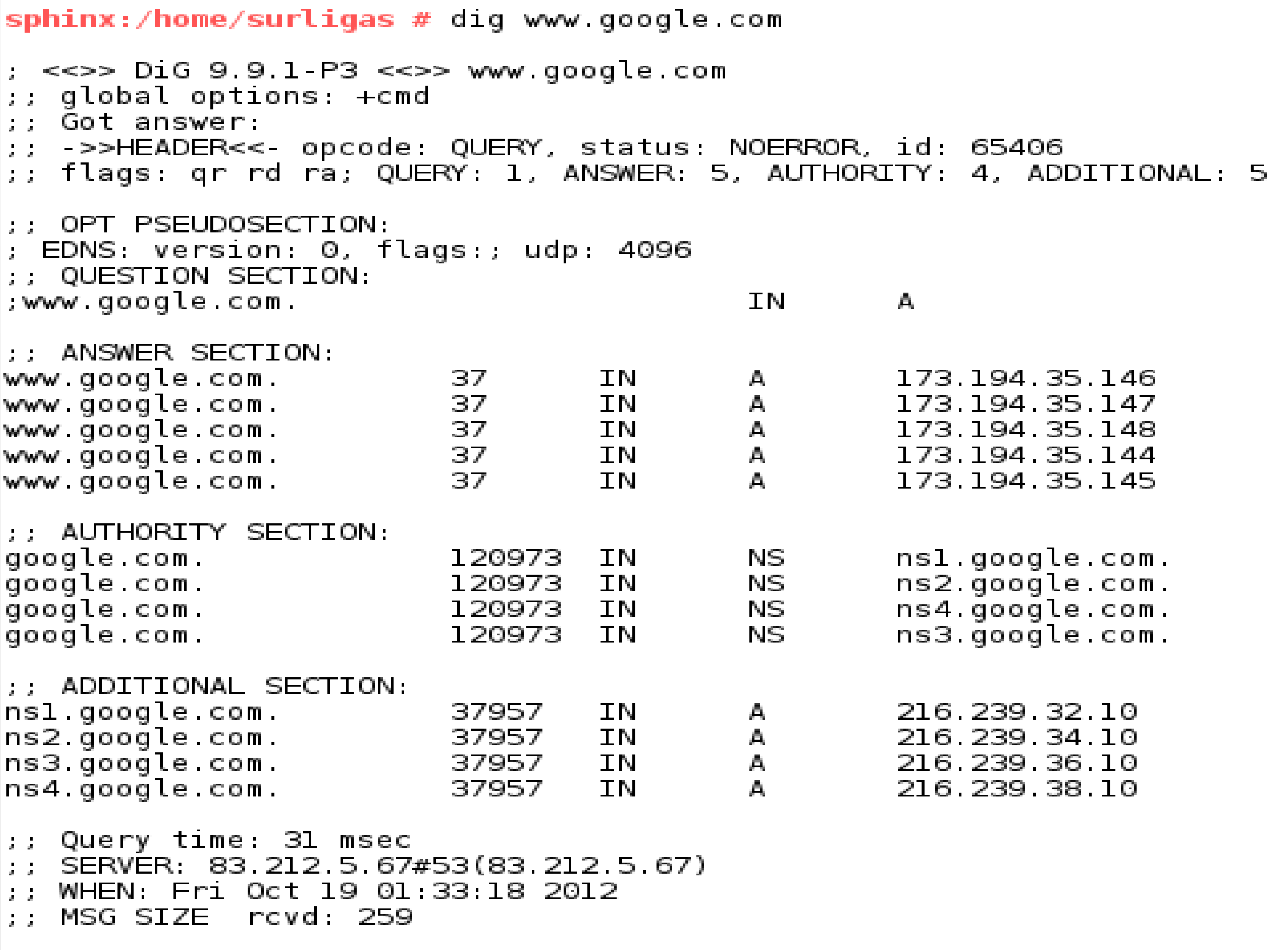


```
mousakas:/home/surligas # traceroute www.ics.forth.gr
traceroute to www.ics.forth.gr (139.91.151.170), 30 hops max, 40 byte packets using UDP
 1  139.91.68.253 (139.91.68.253)  3.141 ms   2.077 ms   0.984 ms
 2  139.91.34.85 (139.91.34.85)  0.631 ms   0.591 ms   0.652 ms
 3  www.ics.forth.gr (139.91.151.170)(H!)  0.527 ms (H!)  0.560 ms (H!)  0.627 ms
```

Dig

- Dig is a tool for debugging the DNS
- It can provide several info like:
  - Which DNS server we used
  - How much time took in order to get the IP from a URL
  - Which is the IP of a URL
  - Which DNS servers were queried
- Very useful for selecting appropriate DNS servers for our network

Dig



```
sphinx:/home/surligas # dig www.google.com

; <<>> DiG 9.9.1-P3 <<>> www.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 65406
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 4, ADDITIONAL: 5

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.google.com.                        IN      A

;; ANSWER SECTION:
www.google.com.         37      IN      A       173.194.35.146
www.google.com.         37      IN      A       173.194.35.147
www.google.com.         37      IN      A       173.194.35.148
www.google.com.         37      IN      A       173.194.35.144
www.google.com.         37      IN      A       173.194.35.145

;; AUTHORITY SECTION:
google.com.             120973  IN      NS      ns1.google.com.
google.com.             120973  IN      NS      ns2.google.com.
google.com.             120973  IN      NS      ns4.google.com.
google.com.             120973  IN      NS      ns3.google.com.

;; ADDITIONAL SECTION:
ns1.google.com.         37957   IN      A       216.239.32.10
ns2.google.com.         37957   IN      A       216.239.34.10
ns3.google.com.         37957   IN      A       216.239.36.10
ns4.google.com.         37957   IN      A       216.239.38.10

;; Query time: 31 msec
;; SERVER: 83.212.5.67#53(83.212.5.67)
;; WHEN: Fri Oct 19 01:33:18 2012
;; MSG SIZE  rcvd: 259
```

GeoIP

- Geoip is a set of tools that reports back to the user, in which county an IP belongs

- Two tools used in combination
  - geoipupdate: Updates the database with countries and IP ranges

  - geoiplookup: Performs the seach. Instead of IP you can use also and a domain name

Questions