Tutorial 1: Ping, Traceroute, Wireshark

TA: Katerina Lionta (klionta@csd.uoc.gr)

cs-335a

Topics

- Background
- Ping
- Traceroute (Tracert)
- Wireshark

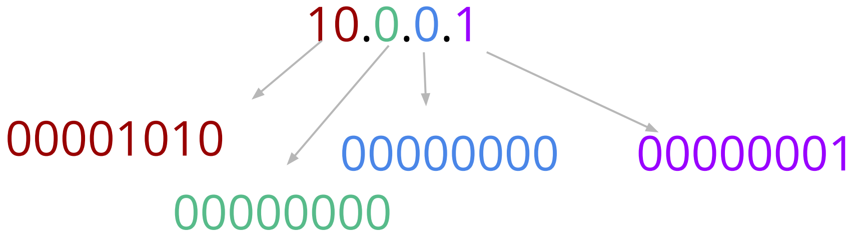
Internet: A global network

The internet is a global network
of routers.
INTERNET
WEB SERVER

- Internet = Courier Company
- We send a message (a packet)
- Tracking Number
    - Steps of our shipment

IP Address

- A network address
- Identifies a network device
- Two versions:
  - ipv4, 32 bits
  - ipv6, 128 bits
- IPv4
  - Decimal representation
  - 4 blocks of 8 bits



10.0.0.1

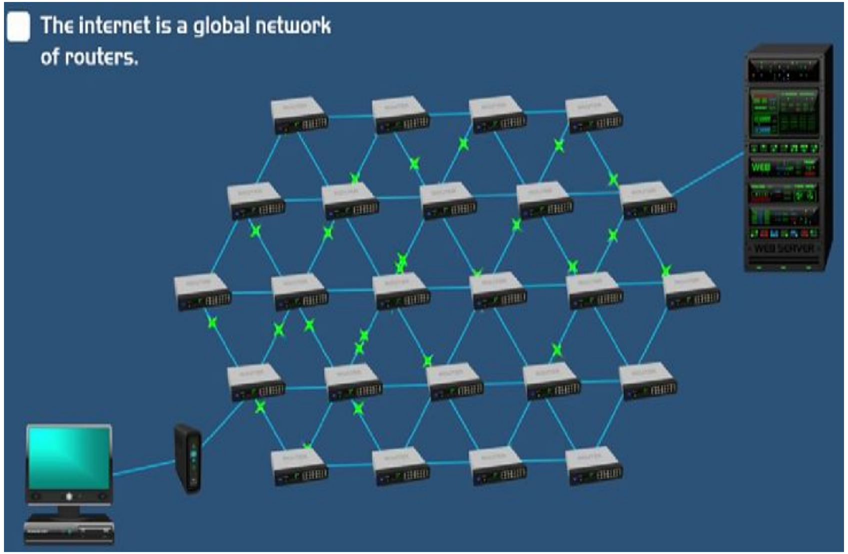
00001010

00000000

00000000

00000001

Routers



- Network devices
- Many interfaces each one with an IP
- They connect with other routers
    - hops (steps of our shipment)
- Pass packets

Use Cases of Ping, Traceroute, Wireshark

- We want to know:
  - The path that our packet follows
  - The time spent at each hop (step of our shipment)
  - The total time it took for our packet (message) to reach the destination

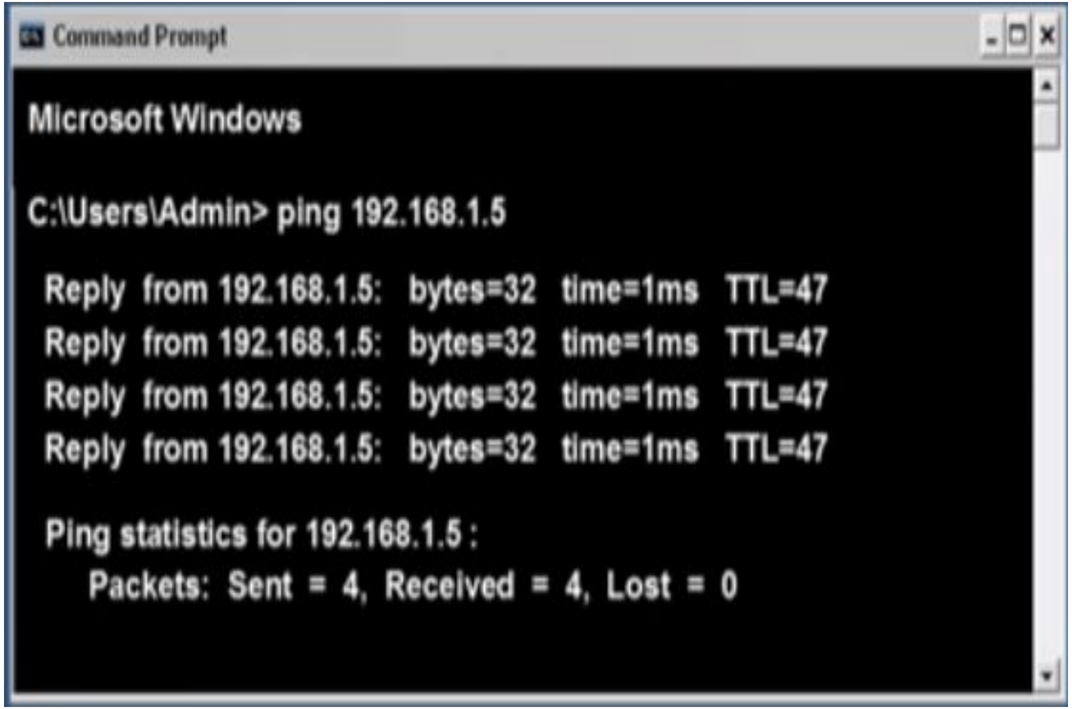
Ping

https://youtu.be/IlicPE38O-s?si=yQLviipmIXEqaFVt

Ping

- A tool for troubleshooting network issues, such as:
  - Test network connectivity (local network, Internet)
  - Test network Interface card
- Available in Windows, Linux, MacOS
- The sender sends packets to the destination and waits for reply
  - 4 packets in windows
  - Not specified in Linux Ctrl+C to stop the running
- **RTT(Round-Trip-Time)**: the time between the transmission of a packet from the transmitter until the reply from the receiver returns
- **Use the flag -4 for IPv4**

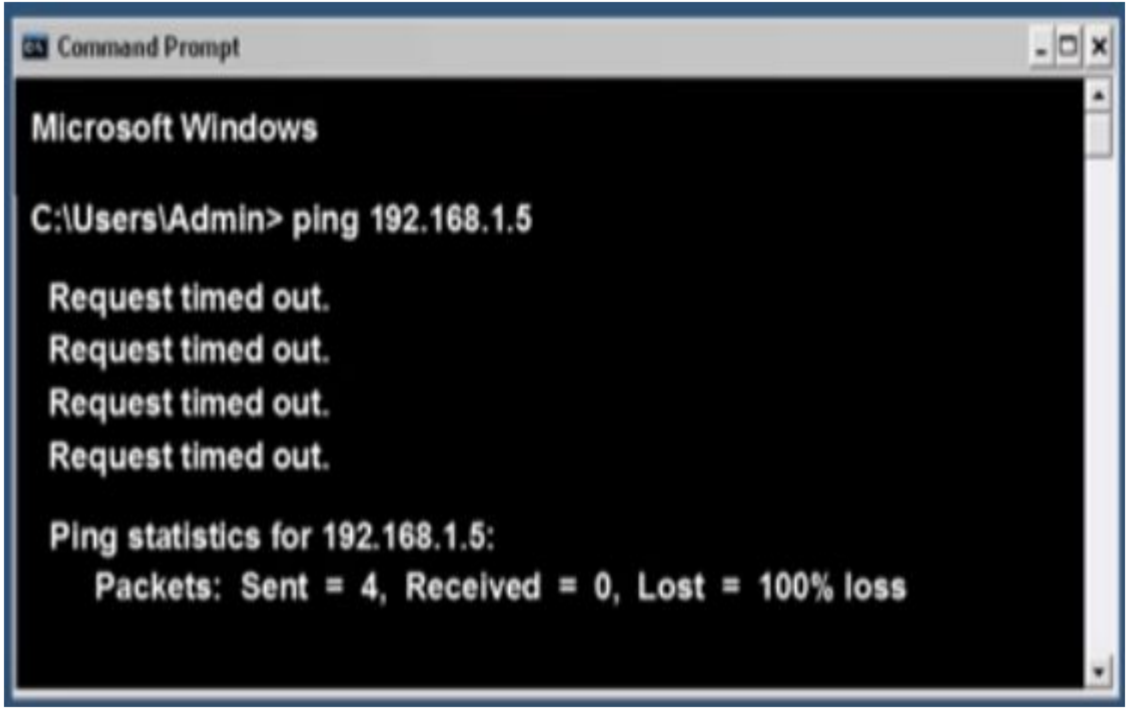
Ping



```
Microsoft Windows

C:\Users\Admin> ping 192.168.1.5

 Reply  from 192.168.1.5:   bytes=32   time=1ms   TTL=47
 Reply  from 192.168.1.5:   bytes=32   time=1ms   TTL=47
 Reply  from 192.168.1.5:   bytes=32   time=1ms   TTL=47
 Reply  from 192.168.1.5:   bytes=32   time=1ms   TTL=47

 Ping statistics for 192.168.1.5 :
    Packets:  Sent  =  4,  Received  =  4,  Lost  =  0
```

- There is connectivity between two hosts, without packet losses
- 4 packets were sent, 4 replies were received
- You can ping an IP address or a server name

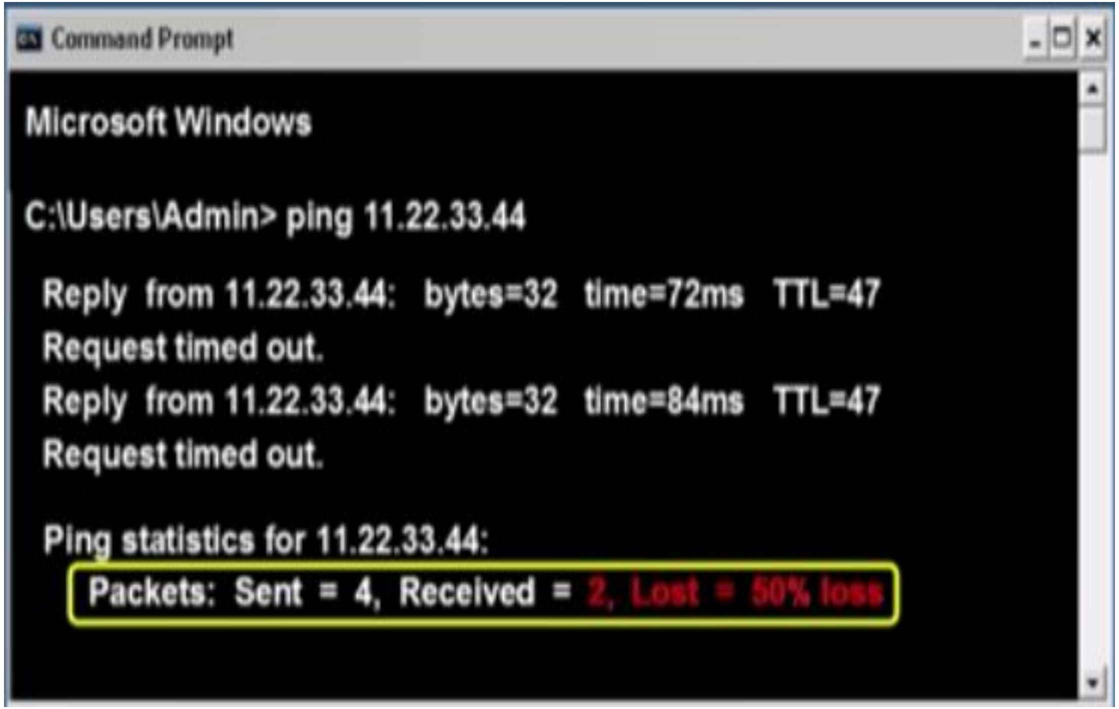
Ping



```
Microsoft Windows

C:\Users\Admin> ping 192.168.1.5

 Request timed out.
 Request timed out.
 Request timed out.
 Request timed out.

 Ping statistics for 192.168.1.5:
     Packets:  Sent  =  4,  Received  =  0,  Lost  =  100% loss
```

- No reply
- There isn’t connectivity between two hosts
- 4 packets were sent, 0 packets were received
- Possible reasons:
    - The receiver is power down
    - Firewall

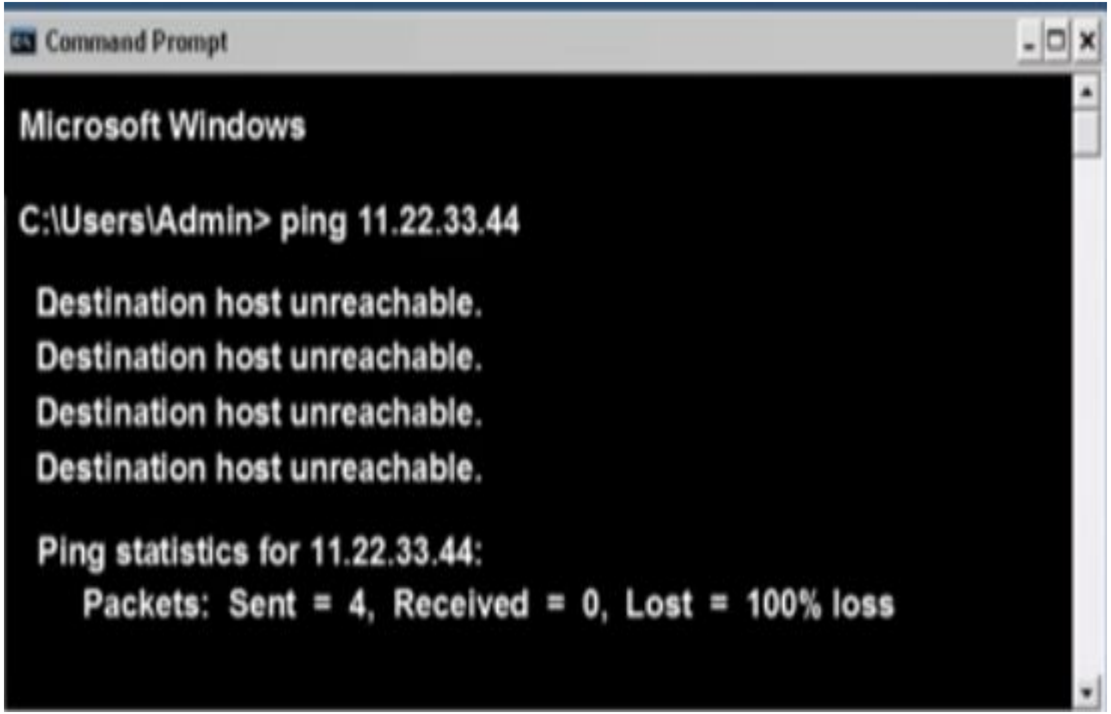
Ping



```
Microsoft Windows

C:\Users\Admin> ping 11.22.33.44

  Reply  from 11.22.33.44:   bytes=32   time=72ms   TTL=47
  Request timed out.
  Reply  from 11.22.33.44:   bytes=32   time=84ms   TTL=47
  Request timed out.

  Ping statistics for 11.22.33.44:
     Packets:  Sent  =  4,  Received  =  2,  Lost  =  50% loss
```

- Not all the data packets reply back to the sender
- **Packet loss**
- Possible reasons:
  - Network congestion
  - Faulty hardware (cables, wiring, network card, modem)

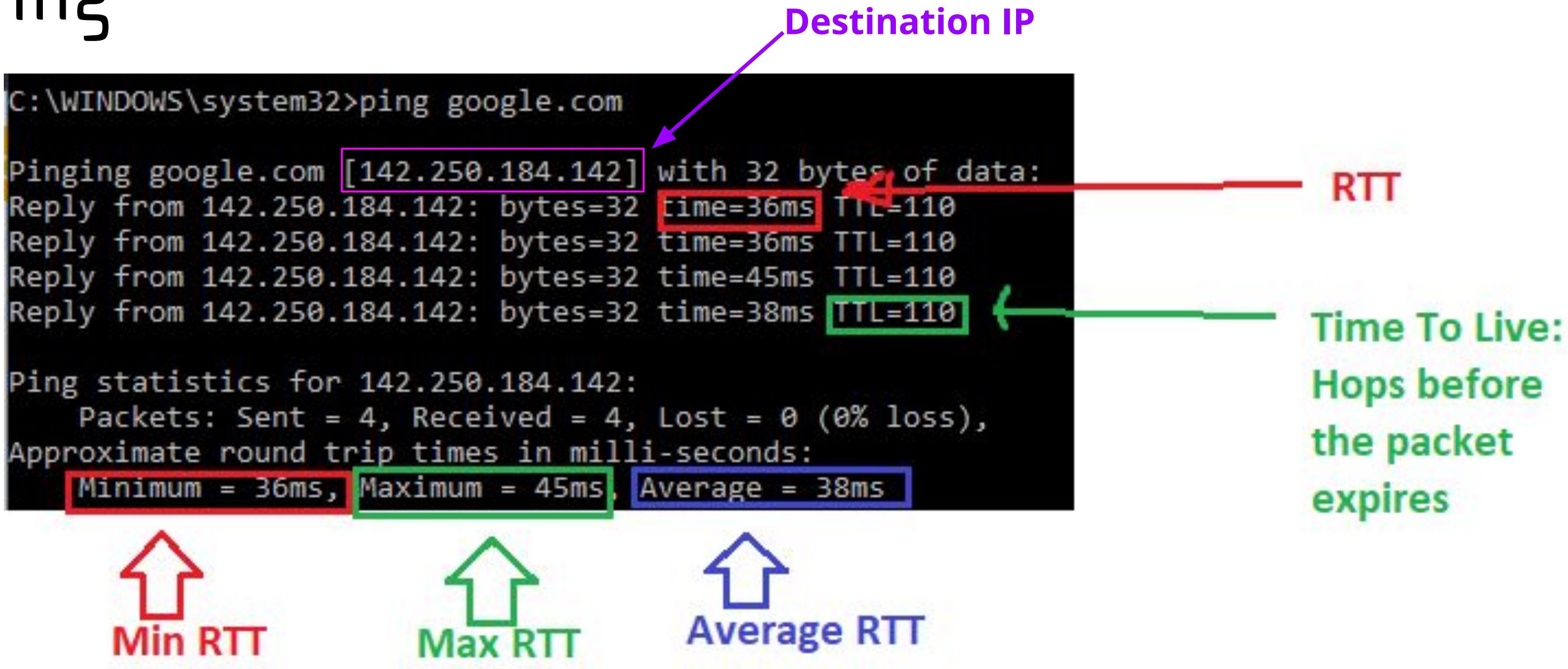
Ping



```
Microsoft Windows

C:\Users\Admin> ping 11.22.33.44

 Destination host unreachable.
 Destination host unreachable.
 Destination host unreachable.
 Destination host unreachable.

 Ping statistics for 11.22.33.44:
     Packets: Sent = 4, Received = 0, Lost = 100% loss
```

- The route to the destination cannot be found
- A router doesn’t have any information on how to route data to the destination
- The destination is disconnecting from the Internet

Ping



```
C:\WINDOWS\system32>ping google.com

Pinging google.com [142.250.184.142] with 32 bytes of data:
Reply from 142.250.184.142: bytes=32 time=36ms TTL=110
Reply from 142.250.184.142: bytes=32 time=36ms TTL=110
Reply from 142.250.184.142: bytes=32 time=45ms TTL=110
Reply from 142.250.184.142: bytes=32 time=38ms TTL=110

Ping statistics for 142.250.184.142:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 36ms, Maximum = 45ms, Average = 38ms
```

Traceroute (tracert)

https://youtu.be/up3bcBLZS74?si=3zoCpaYqv6qA1ALu

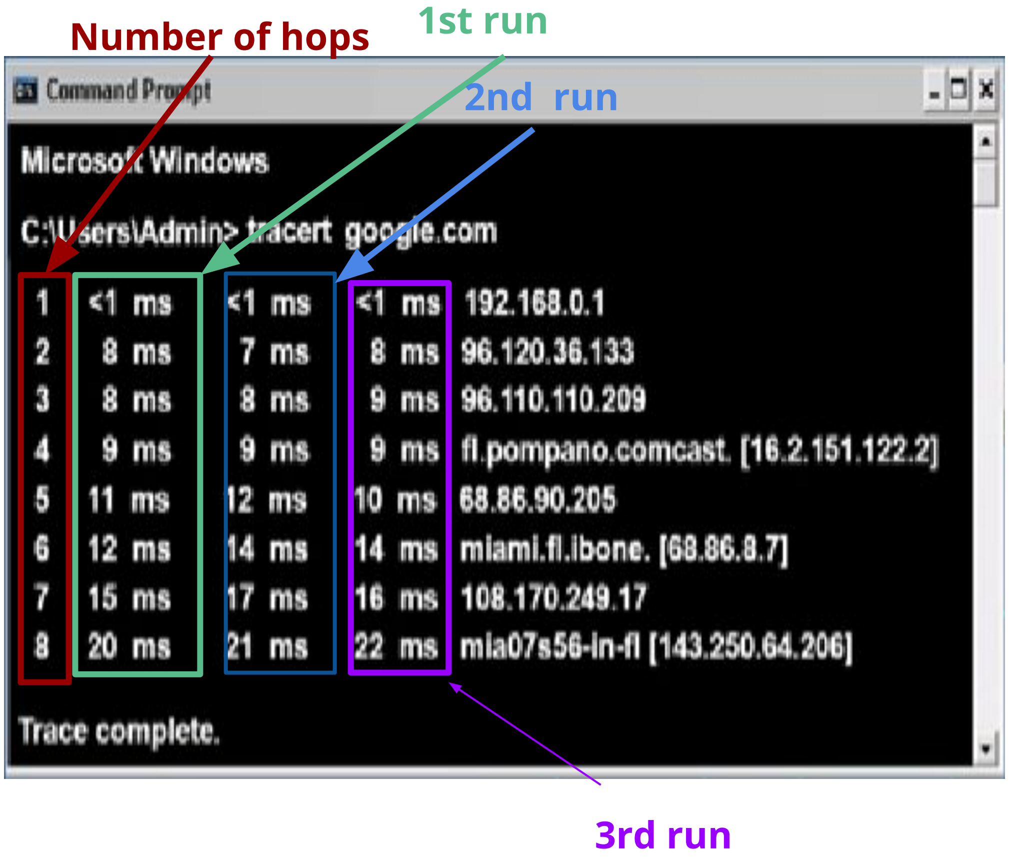
Traceroute

- Used to show the route that the data packets take
- A tool that is used to find the exact path a data packet is taken from the sender to the receiver
- Available in Windows (tracert), UNIX and MacOS
- Pings every router in the path
- Sends packets and measures the RTTs that the data packet took from each router and the destination
- **Use the flag -4 for IPv4**

Traceroute

- **TTL(Time-to-Live)**: how long the packet can live before it discarded, the maximum number of hops that packet can cross until it returns
- Every time a packet passes through a router, the router decreases the TTL by 1
- If TTL=0, the router drops the packet and a reply is transmitted to the sender that identifies the router

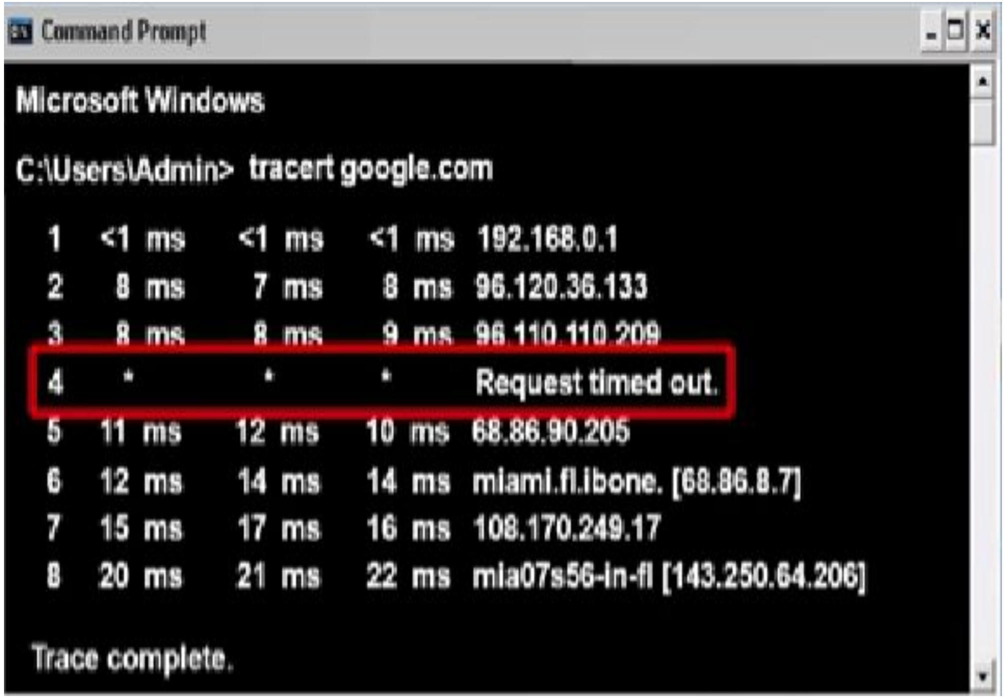
Traceroute



```
Microsoft Windows

C:\Users\Admin> tracert google.com

1   <1 ms   <1 ms   <1 ms  192.168.0.1
2    8 ms    7 ms    8 ms  96.120.36.133
3    8 ms    8 ms    9 ms  96.110.110.209
4    9 ms    9 ms    9 ms  fl.pompano.comcast. [16.2.151.122.2]
5   11 ms   12 ms   10 ms  68.86.90.205
6   12 ms   14 ms   14 ms  miami.fl.ibone. [68.86.8.7]
7   15 ms   17 ms   16 ms  108.170.249.17
8   20 ms   21 ms   22 ms  mia07s56-in-fl [143.250.64.206]

Trace complete.
```

- The RTT of each of three packets from the sender to each router and the destination

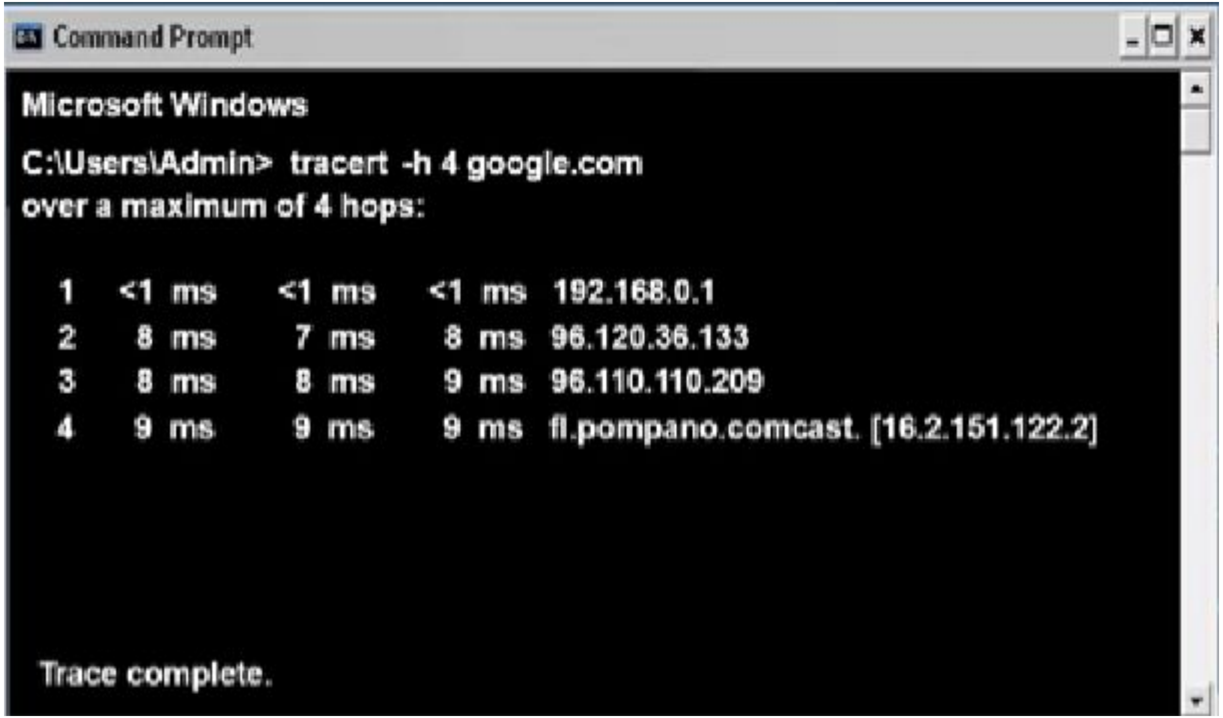
Traceroute



```
Microsoft Windows

C:\Users\Admin> tracert google.com

  1   <1  ms    <1  ms    <1  ms  192.168.0.1
  2    8  ms     7  ms     8  ms  96.120.36.133
  3    8  ms     8  ms     9  ms  96.110.110.209
  4    *           *         *     Request timed out.
  5   11  ms    12  ms    10  ms  68.86.90.205
  6   12  ms    14  ms    14  ms  miami.fl.ibone. [68.86.8.7]
  7   15  ms    17  ms    16  ms  108.170.249.17
  8   20  ms    21  ms    22  ms  mia07s56-in-fl [143.250.64.206]

 Trace complete.
```

- possible reasons:
  - problem with the specific router
  - not configured to return traceroute replies
- The packets passed to the next router

Traceroute



```
Microsoft Windows
C:\Users\Admin>  tracert  -h 4 google.com
over a maximum of 4 hops:

   1    <1  ms     <1  ms     <1  ms   192.168.0.1
   2     8  ms      7  ms      8  ms   96.120.36.133
   3     8  ms      8  ms      9  ms   96.110.110.209
   4     9  ms      9  ms      9  ms   fl.pompano.comcast. [16.2.151.122.2]

 Trace complete.
```

- Sets the TTL=4
- When the packet traverses 4 hops,it is dropped

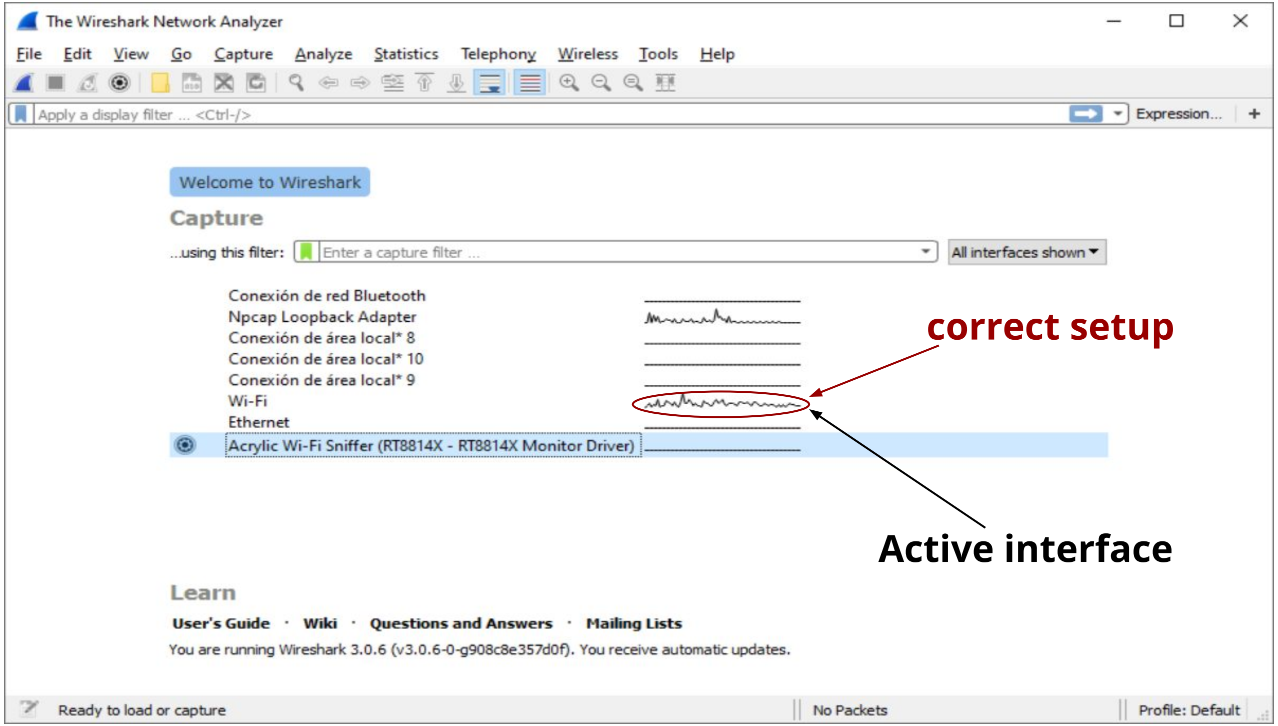
How can we do that?

Wireshark

Wireshark

- Packet sniffer: tool used for capturing network packets
- Download:
    - Windows and Macos:
        - https://www.wireshark.org/download.html
        - In windows run the Wireshark as administrator
    - Linux:
        - https://linuxhint.com/install_configure_wireshark_ubuntu/

Wireshark: Setup



Capture

...using this filter:

Conexión de red Bluetooth
Npcap Loopback Adapter
Conexión de área local* 8
Conexión de área local* 10
Conexión de área local* 9
Wi-Fi
Ethernet
Acrylic Wi-Fi Sniffer (RT8814X - RT8814X Monitor Driver)

correct setup

Active interface

Learn

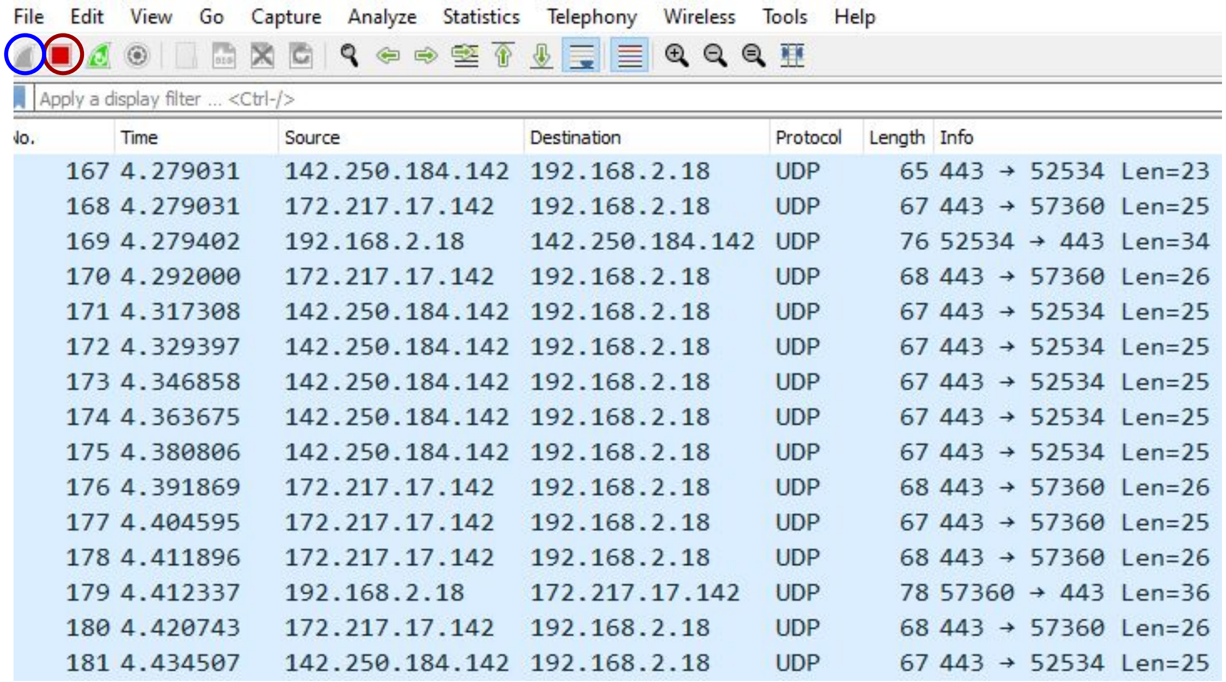
User's Guide · Wiki · Questions and Answers · Mailing Lists

You are running Wireshark 3.0.6 (v3.0.6-0-g908c8e357d0f). You receive automatic updates.

Ready to load or capture | No Packets | Profile: Default

- **Run** the Wireshark **as administrator**
- Select an active interface

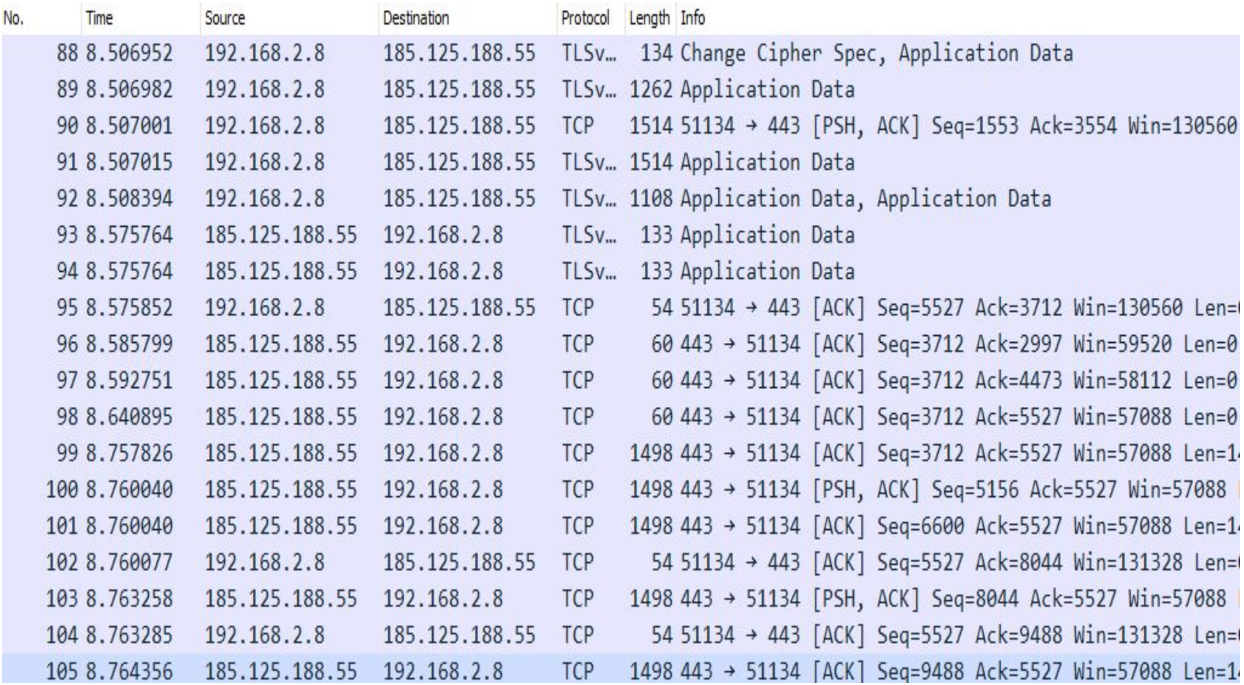
Wireshark: Capturing



```
lo.     Time       Source           Destination      Protocol  Length  Info
    167 4.279031   142.250.184.142  192.168.2.18     UDP           65  443 → 52534  Len=23
    168 4.279031   172.217.17.142   192.168.2.18     UDP           67  443 → 57360  Len=25
    169 4.279402   192.168.2.18     142.250.184.142  UDP           76  52534 → 443  Len=34
    170 4.292000   172.217.17.142   192.168.2.18     UDP           68  443 → 57360  Len=26
    171 4.317308   142.250.184.142  192.168.2.18     UDP           67  443 → 52534  Len=25
    172 4.329397   142.250.184.142  192.168.2.18     UDP           67  443 → 52534  Len=25
    173 4.346858   142.250.184.142  192.168.2.18     UDP           67  443 → 52534  Len=25
    174 4.363675   142.250.184.142  192.168.2.18     UDP           67  443 → 52534  Len=25
    175 4.380806   142.250.184.142  192.168.2.18     UDP           67  443 → 52534  Len=25
    176 4.391869   172.217.17.142   192.168.2.18     UDP           68  443 → 57360  Len=26
    177 4.404595   172.217.17.142   192.168.2.18     UDP           67  443 → 57360  Len=25
    178 4.411896   172.217.17.142   192.168.2.18     UDP           68  443 → 57360  Len=26
    179 4.412337   192.168.2.18     172.217.17.142   UDP           78  57360 → 443  Len=36
    180 4.420743   172.217.17.142   192.168.2.18     UDP           68  443 → 57360  Len=26
    181 4.434507   142.250.184.142  192.168.2.18     UDP           67  443 → 52534  Len=25
```

- Start capturing packets
- Stop capturing 🟥

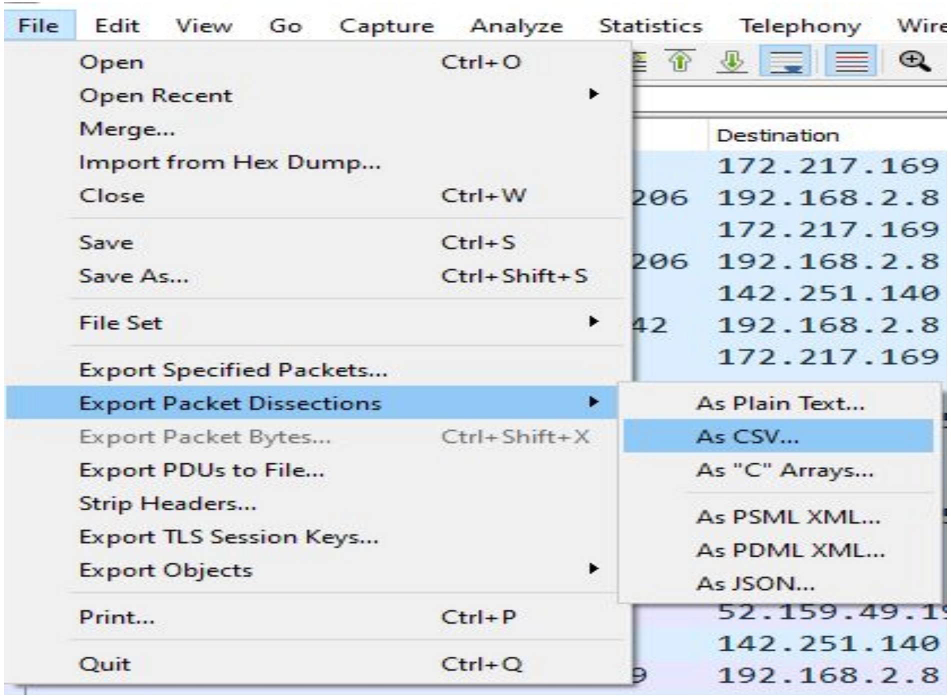
Wireshark



No.	Time	Source	Destination	Protocol	Length	Info
88	8.506952	192.168.2.8	185.125.188.55	TLSv…	134	Change Cipher Spec, Application Data
89	8.506982	192.168.2.8	185.125.188.55	TLSv…	1262	Application Data
90	8.507001	192.168.2.8	185.125.188.55	TCP	1514	51134 → 443 [PSH, ACK] Seq=1553 Ack=3554 Win=130560
91	8.507015	192.168.2.8	185.125.188.55	TLSv…	1514	Application Data
92	8.508394	192.168.2.8	185.125.188.55	TLSv…	1108	Application Data, Application Data
93	8.575764	185.125.188.55	192.168.2.8	TLSv…	133	Application Data
94	8.575764	185.125.188.55	192.168.2.8	TLSv…	133	Application Data
95	8.575852	192.168.2.8	185.125.188.55	TCP	54	51134 → 443 [ACK] Seq=5527 Ack=3712 Win=130560 Len=
96	8.585799	185.125.188.55	192.168.2.8	TCP	60	443 → 51134 [ACK] Seq=3712 Ack=2997 Win=59520 Len=0
97	8.592751	185.125.188.55	192.168.2.8	TCP	60	443 → 51134 [ACK] Seq=3712 Ack=4473 Win=58112 Len=0
98	8.640895	185.125.188.55	192.168.2.8	TCP	60	443 → 51134 [ACK] Seq=3712 Ack=5527 Win=57088 Len=0
99	8.757826	185.125.188.55	192.168.2.8	TCP	1498	443 → 51134 [ACK] Seq=3712 Ack=5527 Win=57088 Len=1
100	8.760040	185.125.188.55	192.168.2.8	TCP	1498	443 → 51134 [PSH, ACK] Seq=5156 Ack=5527 Win=57088
101	8.760040	185.125.188.55	192.168.2.8	TCP	1498	443 → 51134 [ACK] Seq=6600 Ack=5527 Win=57088 Len=1
102	8.760077	192.168.2.8	185.125.188.55	TCP	54	51134 → 443 [ACK] Seq=5527 Ack=8044 Win=131328 Len=
103	8.763258	185.125.188.55	192.168.2.8	TCP	1498	443 → 51134 [PSH, ACK] Seq=8044 Ack=5527 Win=57088
104	8.763285	192.168.2.8	185.125.188.55	TCP	54	51134 → 443 [ACK] Seq=5527 Ack=9488 Win=131328 Len=
105	8.764356	185.125.188.55	192.168.2.8	TCP	1498	443 → 51134 [ACK] Seq=9488 Ack=5527 Win=57088 Len=1

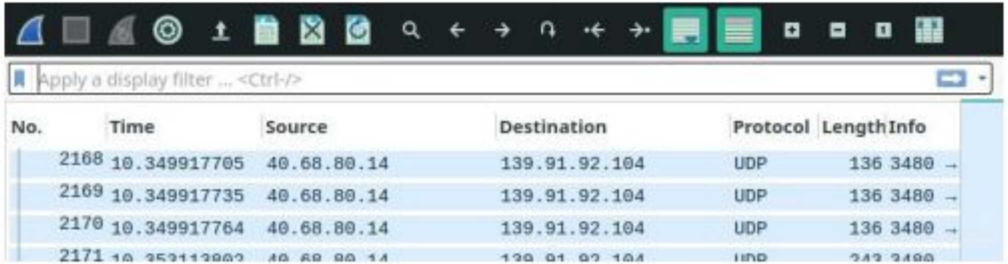
- **No.**: the serial number of the packet
- **Time**: the time of the transmission/receiving of the packet (starts from 0, the moment that the capturing started) in seconds
- **Source**: the source IP address
- **Destination**: the destination IP address
- **Protocol**: the protocol used
- **Length**: the length of the packets in bytes
- **Info**: extra information about the packet (header fields, flags etc)

Wireshark: Export files



- File > Export Packet Dissections > AS CSV…

Wireshark: Filtering



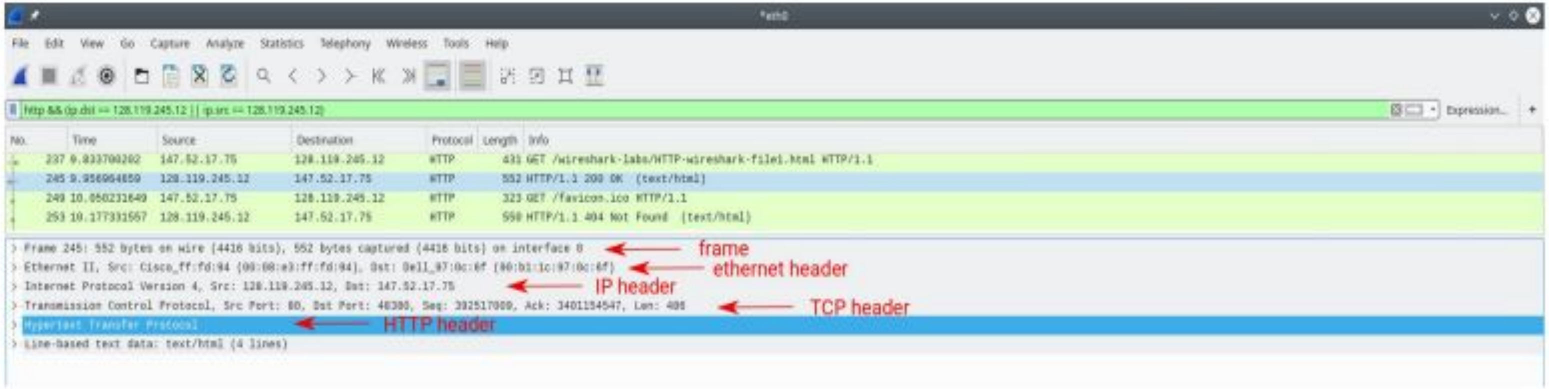
No.	Time	Source	Destination	Protocol	Length	Info
2168	10.349917705	40.68.80.14	139.91.92.104	UDP	136	3480 →
2169	10.349917735	40.68.80.14	139.91.92.104	UDP	136	3480 →
2170	10.349917764	40.68.80.14	139.91.92.104	UDP	136	3480 →
2171	10.353113802	40.68.80.14	139.91.92.104	UDP	243	3480

- You can filter with:
  - Transfer protocol name (tcp,udp etc)
  - Source IP, destination IP (ip.src==192.168.0.0, ip.dst==192.168.0.0)
- You can use logical operators:
  - and,or
  - && ,||,!
- Examples:
  - ip.src != 10.43.54.65 or ip.dst != 10.43.54.65
  - tcp
  - udp

https://wiki.wireshark.org/DisplayFilters

Wireshark: Encapsulation

- Encapsulation allows us to use different protocols in all levels of the TCP/IP stack.
- Wireshark shows us the headers of all these levels (e.g. an HTTP packet)

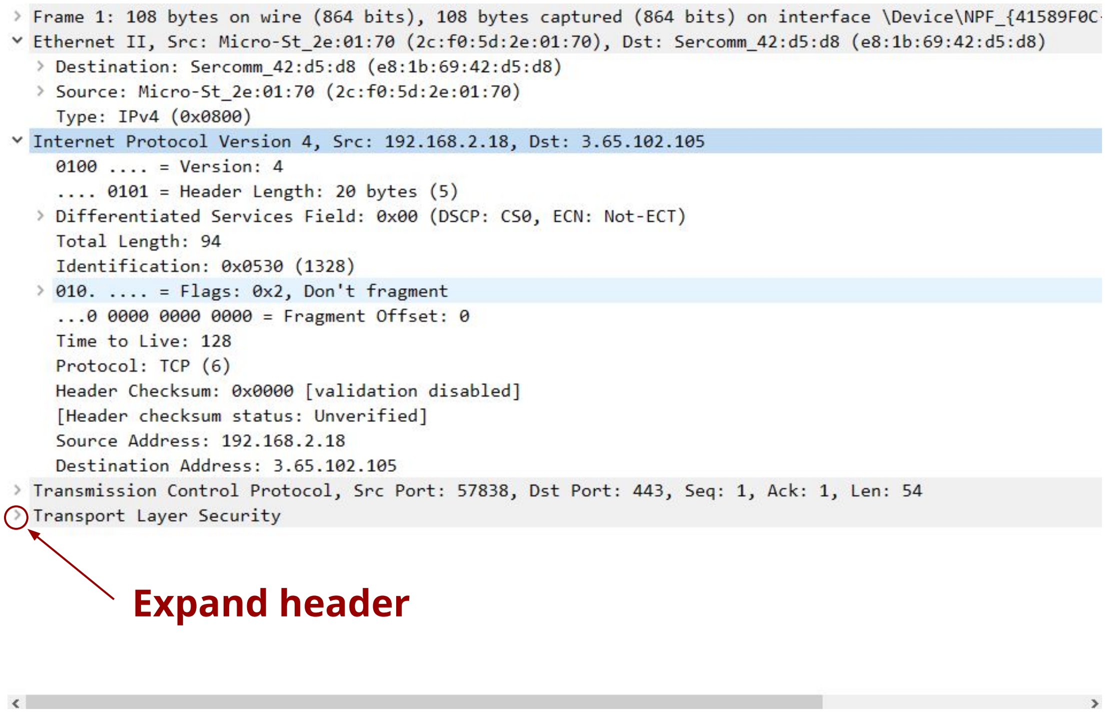

*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression...
frame
ethernet header
IP header
TCP header
HTTP header

```
http && (ip.dst == 128.119.245.12 || ip.src == 128.119.245.12)
```

No.	Time	Source	Destination	Protocol	Length	Info
237	9.833700202	147.52.17.75	128.119.245.12	HTTP	431	GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1
245	9.956964859	128.119.245.12	147.52.17.75	HTTP	552	HTTP/1.1 200 OK (text/html)
249	10.050231649	147.52.17.75	128.119.245.12	HTTP	323	GET /favicon.ico HTTP/1.1
253	10.177331557	128.119.245.12	147.52.17.75	HTTP	550	HTTP/1.1 404 Not Found (text/html)

```
> Frame 245: 552 bytes on wire (4416 bits), 552 bytes captured (4416 bits) on interface 0
> Ethernet II, Src: Cisco_ff:fd:94 (00:08:e3:ff:fd:94), Dst: Dell_07:0c:6f (00:b1:1c:07:0c:6f)
> Internet Protocol Version 4, Src: 128.119.245.12, Dst: 147.52.17.75
> Transmission Control Protocol, Src Port: 80, Dst Port: 40300, Seq: 392517009, Ack: 3401154547, Len: 498
> Hypertext Transfer Protocol
> Line-based text data: text/html (4 lines)
```

Wireshark: Encapsulation



```
> Frame 1: 108 bytes on wire (864 bits), 108 bytes captured (864 bits) on interface \Device\NPF_{41589F0C
v Ethernet II, Src: Micro-St_2e:01:70 (2c:f0:5d:2e:01:70), Dst: Sercomm_42:d5:d8 (e8:1b:69:42:d5:d8)
    > Destination: Sercomm_42:d5:d8 (e8:1b:69:42:d5:d8)
    > Source: Micro-St_2e:01:70 (2c:f0:5d:2e:01:70)
      Type: IPv4 (0x0800)
v Internet Protocol Version 4, Src: 192.168.2.18, Dst: 3.65.102.105
      0100 .... = Version: 4
      .... 0101 = Header Length: 20 bytes (5)
    > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      Total Length: 94
      Identification: 0x0530 (1328)
    > 010. .... = Flags: 0x2, Don't fragment
      ...0 0000 0000 0000 = Fragment Offset: 0
      Time to Live: 128
      Protocol: TCP (6)
      Header Checksum: 0x0000 [validation disabled]
      [Header checksum status: Unverified]
      Source Address: 192.168.2.18
      Destination Address: 3.65.102.105
> Transmission Control Protocol, Src Port: 57838, Dst Port: 443, Seq: 1, Ack: 1, Len: 54
> Transport Layer Security
```

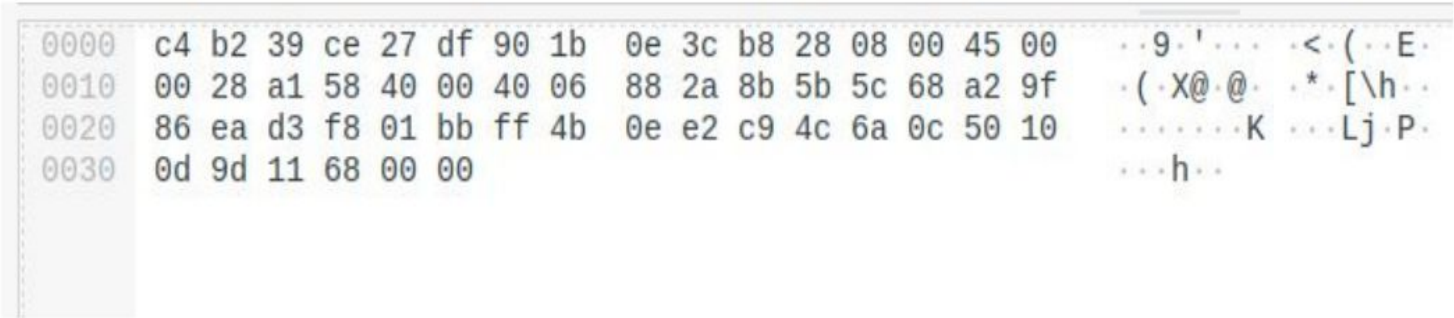
Expand header

- Five headers (Physical layer, Link layer, Network layer, Transport layer, App layer)

Wireshark: Packet details

As mentioned, you can see the details of each package by clicking on it. You can double-click to open it in a new window. There you can see the packet's details, as they're shown on the previous slide.

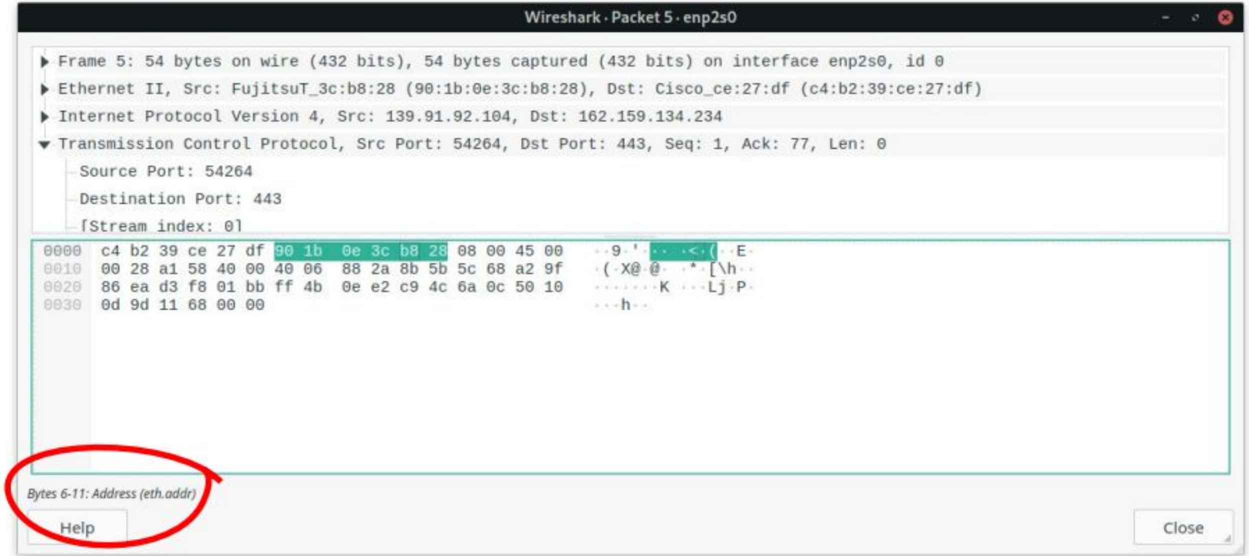
You can also see the hex format of the packet:



```
0000  c4 b2 39 ce 27 df 90 1b  0e 3c b8 28 08 00 45 00   ··9·'···  ·<·(··E·
0010  00 28 a1 58 40 00 40 06  88 2a 8b 5b 5c 68 a2 9f   ·(·X@·@·  ·*·[\h··
0020  86 ea d3 f8 01 bb ff 4b  0e e2 c9 4c 6a 0c 50 10   ·······K  ···Lj·P·
0030  0d 9d 11 68 00 00                                  ···h··
```

Wireshark: Packet details

You can hover over the bytes and see what they represent (see bottom left corner):



```
▶ Frame 5: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface enp2s0, id 0
▶ Ethernet II, Src: FujitsuT_3c:b8:28 (90:1b:0e:3c:b8:28), Dst: Cisco_ce:27:df (c4:b2:39:ce:27:df)
▶ Internet Protocol Version 4, Src: 139.91.92.104, Dst: 162.159.134.234
▼ Transmission Control Protocol, Src Port: 54264, Dst Port: 443, Seq: 1, Ack: 77, Len: 0
    Source Port: 54264
    Destination Port: 443
    [Stream index: 0]

0000  c4 b2 39 ce 27 df 90 1b  0e 3c b8 28 08 00 45 00   ··9·'···  ·<·(··E·
0010  00 28 a1 58 40 00 40 06  88 2a 8b 5b 5c 68 a2 9f   ·(·X@·@·  ·*·[\h··
0020  86 ea d3 f8 01 bb ff 4b  0e e2 c9 4c 6a 0c 50 10   ·······K  ···Lj·P·
0030  0d 9d 11 68 00 00                                  ···h··
```

Bytes 6-11: Address (eth.addr)

Thank You