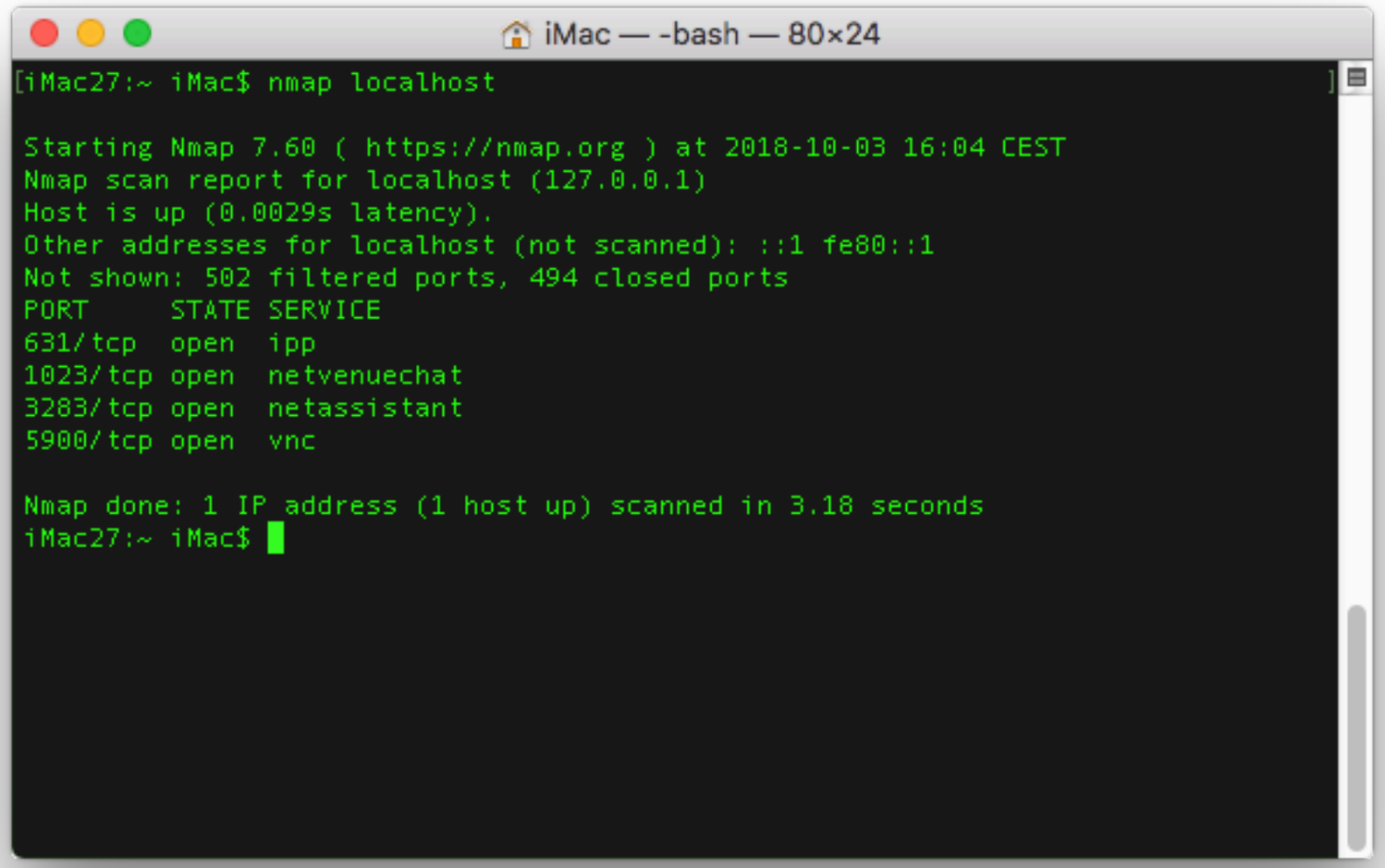
Wireshark

Les 4 04/10/2018

M. DIMA

Poorten

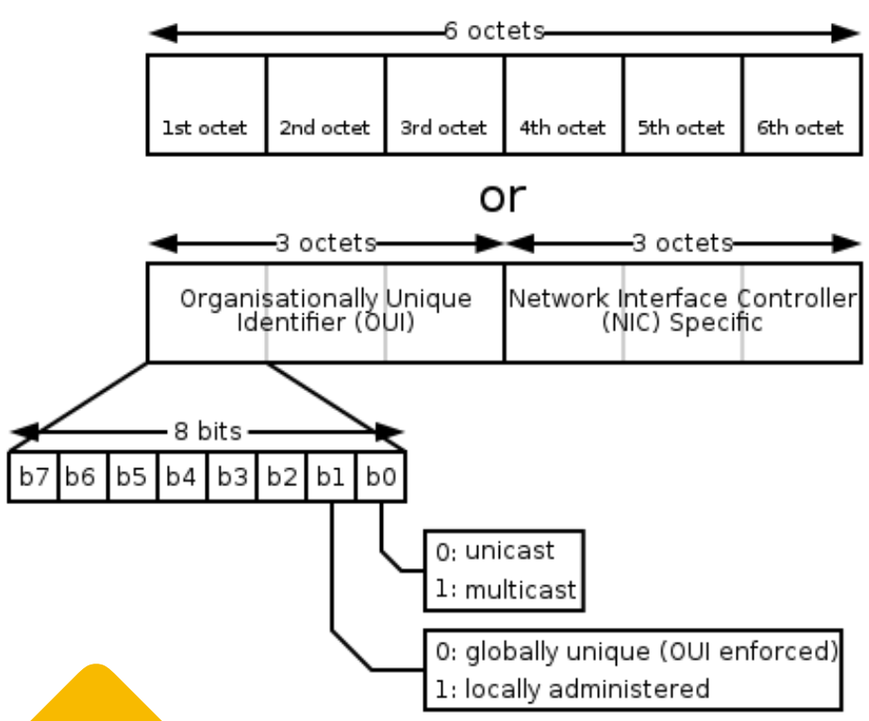
- nmap localhost

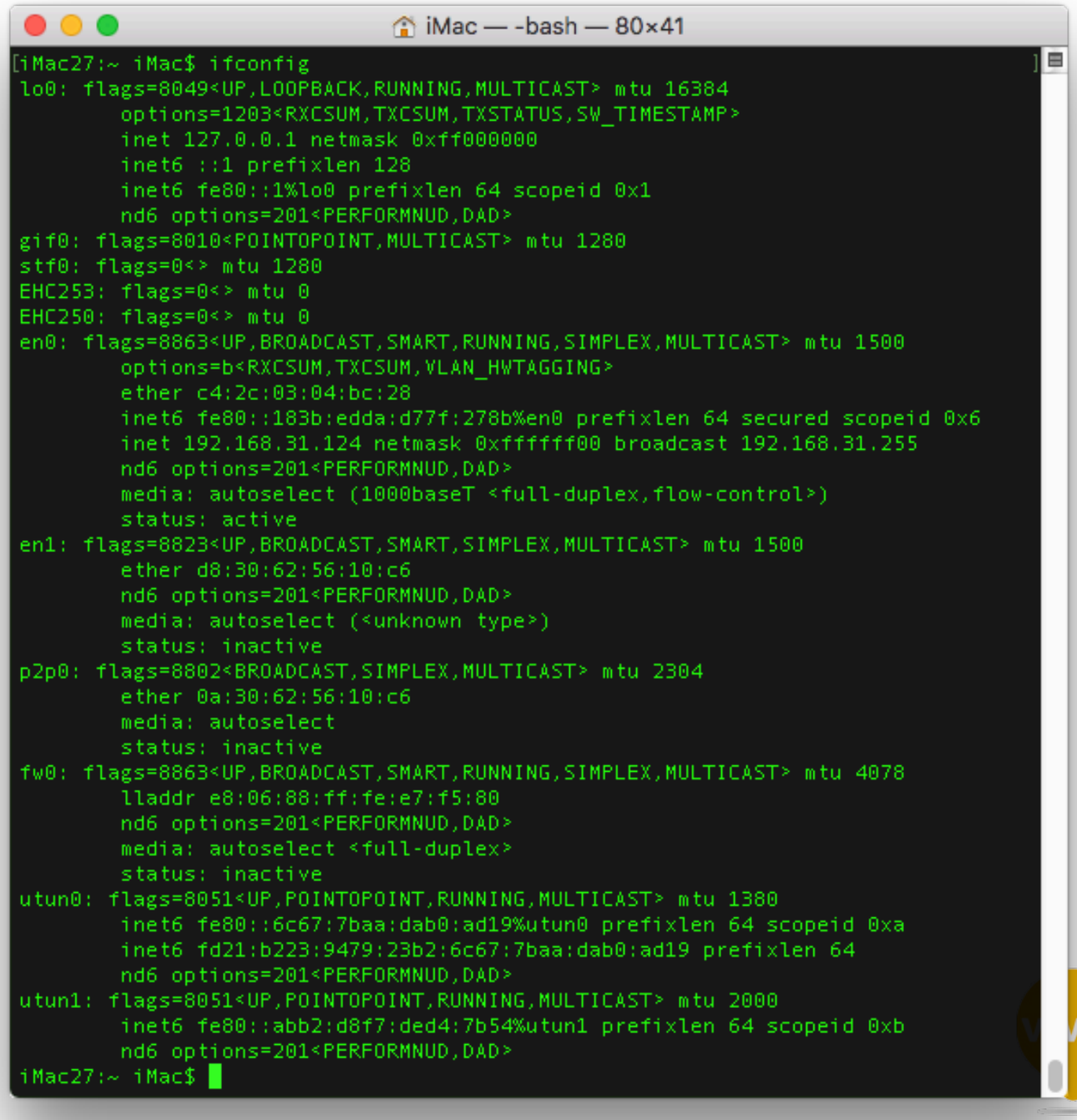


MAC Adres

- Media Access Control address 00:0C:6E:D2:11:E6

- Broadcast FF:FF:FF:FF:FF:FF





Protocol

- Set voorgedefinieerde regels over hoe iets moet gebeuren bv TCP-IP **T**ransmission **C**ontrol **P**rotocol / **I**nternet **P**rotocol

	OSI Layer	TCP/IP	Datagrams are called
Software	**Layer 7** Application	HTTP, SMTP, IMAP, SNMP, POP3, FTP	**Upper Layer Data**
	Layer 6 Presentation	ASCII Characters, MPEG, SSL, TSL, Compression (Encryption & Decryption)	
	Layer 5 Session	NetBIOS, SAP, Handshaking connection	
	Layer 4 Transport	TCP, UDP	**Segment**
	Layer 3 Network	IPv4, IPv6, ICMP, IPSec, MPLS, ARP	**Packet**
Hardware	**Layer 2** Data Link	Ethernet, 802.1x, PPP, ATM, Fiber Channel, MPLS, FDDI, MAC Addresses	**Frame**
	Layer 1 Physical	Cables, Connectors, Hubs (DLS, RS232, 10BaseT, 100BaseTX, ISDN, T1)	**Bits**

apps zelf

converters data
app -> network

manages connections

transfer data h -> h + data integriteit

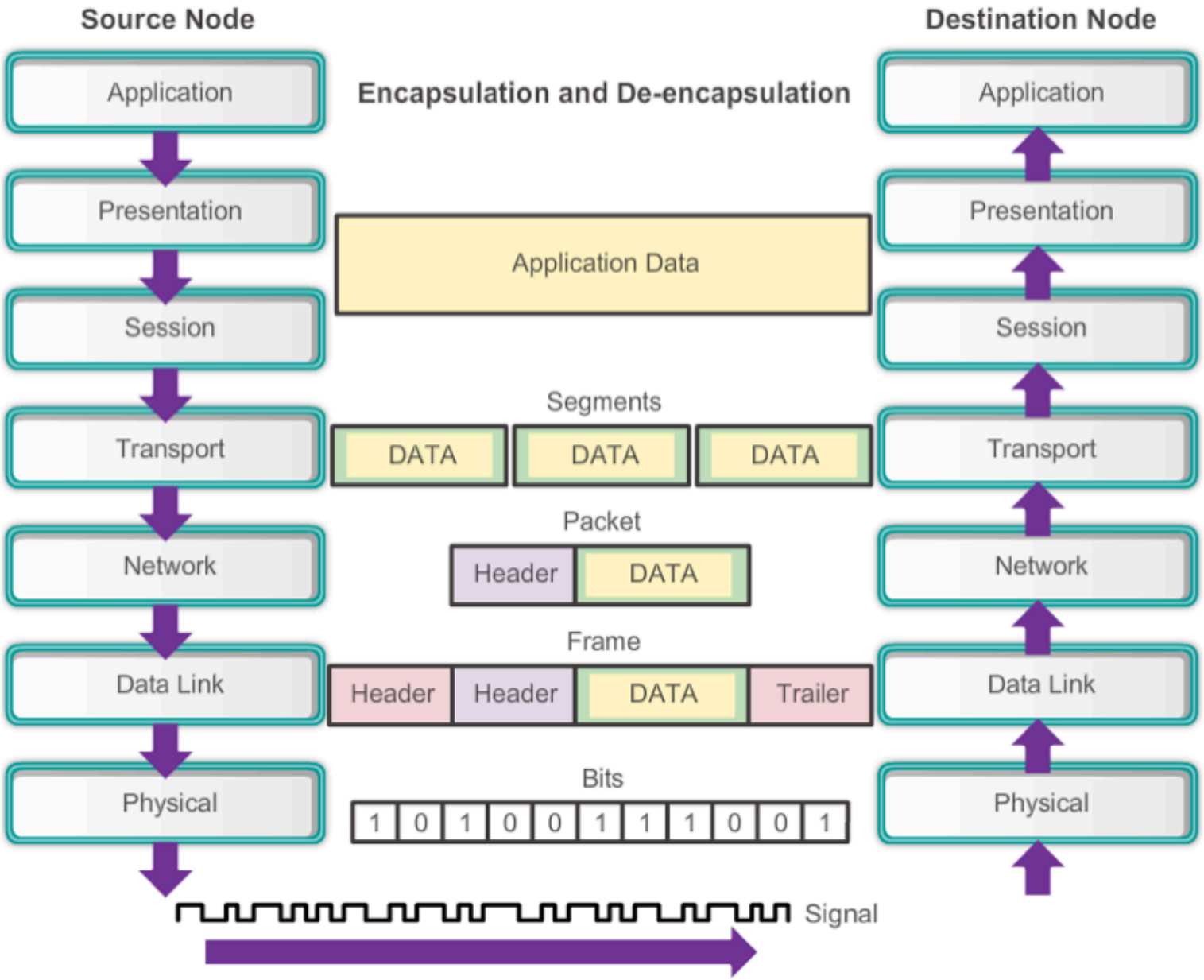
switching & routing

MAC & LLC

Doel van de fysieke laag

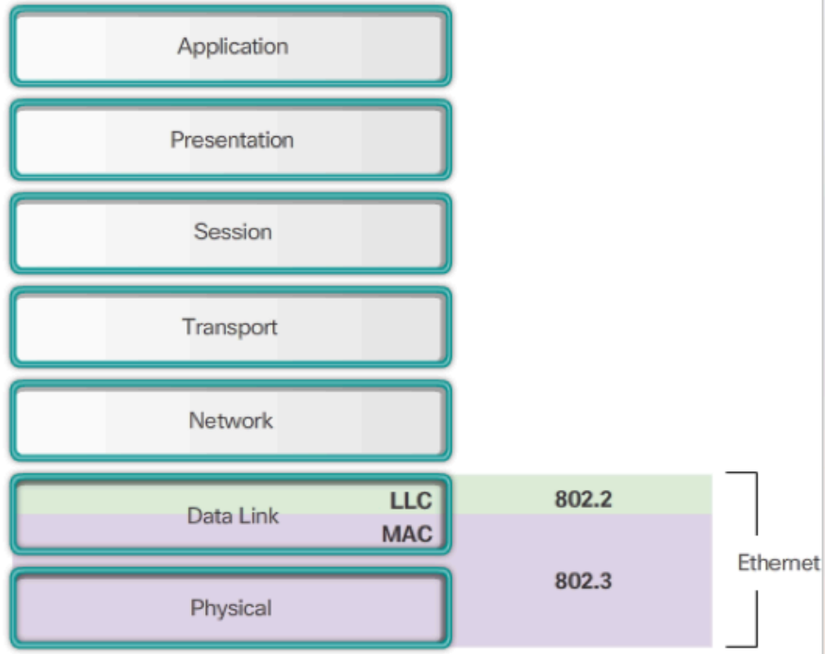
De fysieke laag

De fysieke laag biedt de middelen om de bits van een frame gevormd door de datalinklaag over het netwerk medium te sturen.



Ethernet Operatie

LLC en MAC Sublagen



Ethernet

- Eén van de meest gebruikte LAN-technologieën
- Werkt in de datalinklaag en de fysieke laag
- Familie van netwerktechnologieën die zijn gedefinieerd in de IEEE 802.2 en 802.3 standaarden.
- Ondersteunt data bandbreedtes van 10, 100, 1 000, 10 000, 40 000 en 100 000 Mbps (100 Gbps)

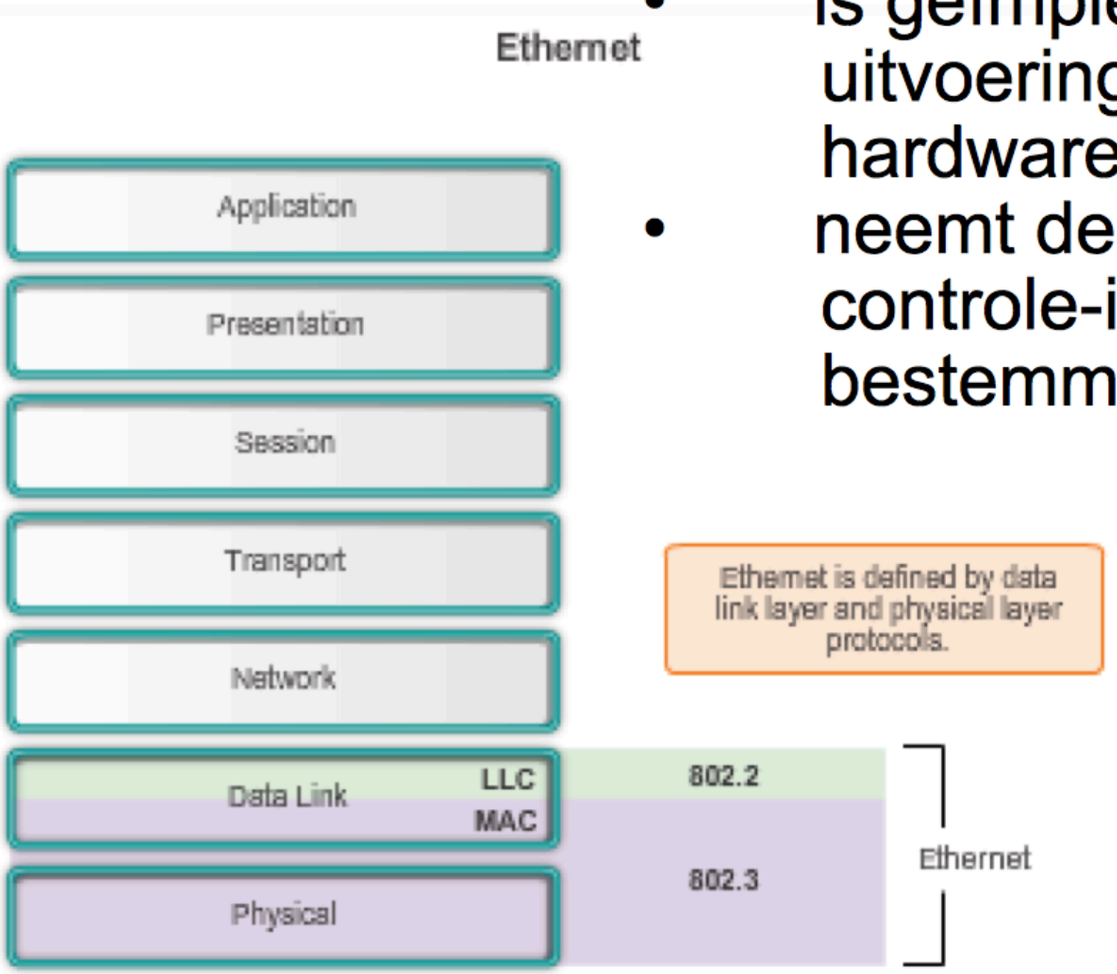
Ethernet standaarden

- Definiëren laag 2 protocollen en laag 1 technologieën.
- Twee afzonderlijke sublagen van de datalinklaag om het te laten werken - **Logical Link Control (LLC)** en de **MAC** sublagen.

LLC en MAC Sublagen (vervolg)

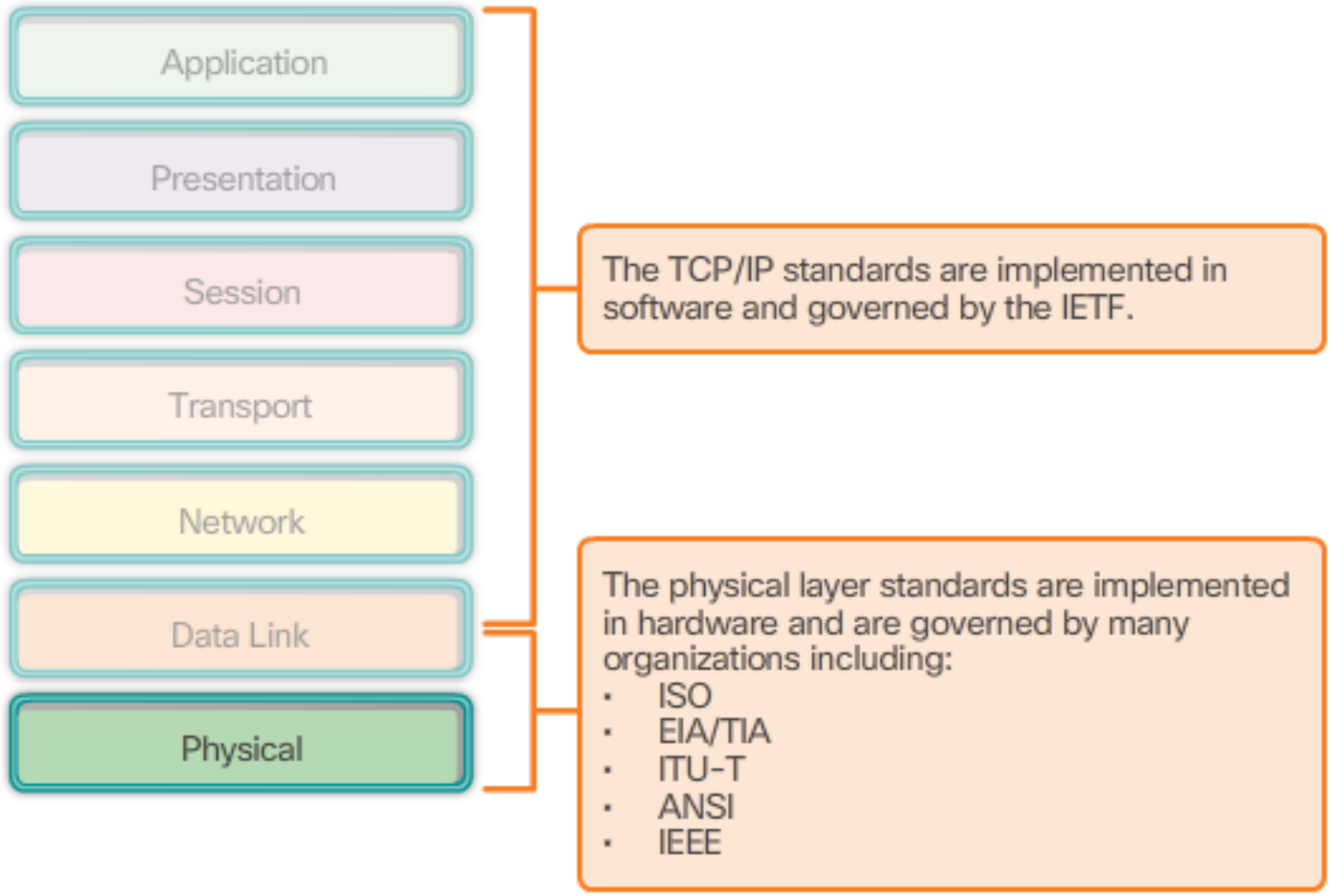
De Ethernet **LLC** sublaag

- verzorgt de communicatie tussen de netwerksoftware en de apparaat hardware
- is geïmplementeerd in de software, en de uitvoering ervan is onafhankelijk van de hardware
- neemt de netwerkprotocolgegevens en voegt controle-informatie toe om het pakket naar de bestemming te helpen



De **MAC** sublaag vormt de onderste sublaag van de datalink-laag. MAC wordt uitgevoerd door de hardware, meestal in de computer NIC. De details worden gespecificeerd in de IEEE 802.3-normen.

Standaarden van de fysieke laag

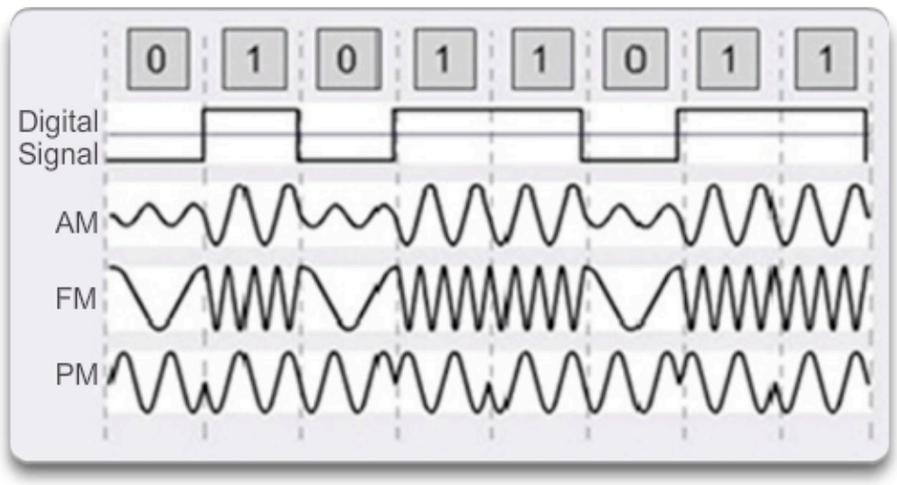


Doel van de fysieke laag

Het fysieke medium

Electrical Signals -
Copper cable

Light Pulse -
Fiber-optic cable



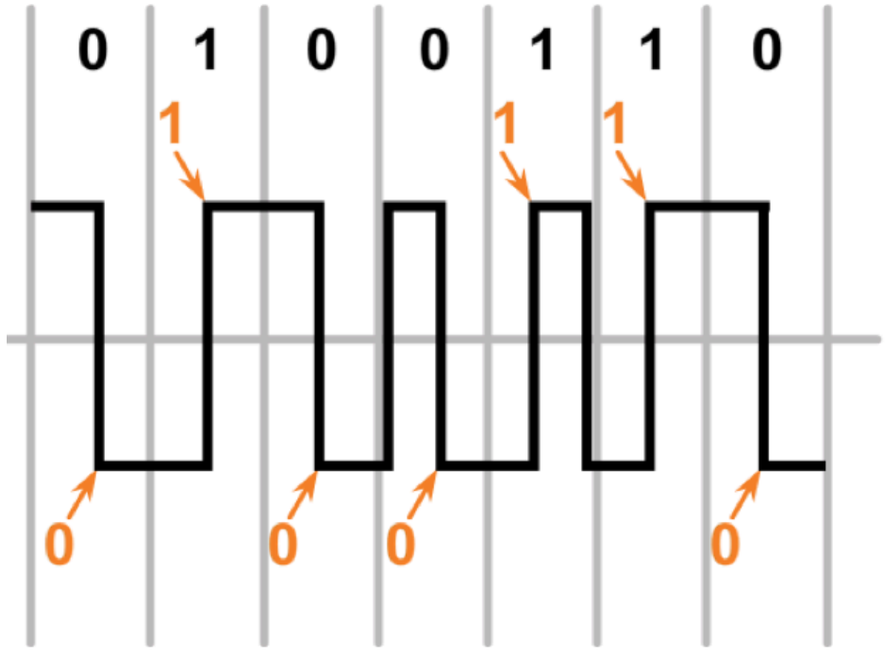
Microwave Signals -
Wireless

Fundamentele principes van de fysieke laag

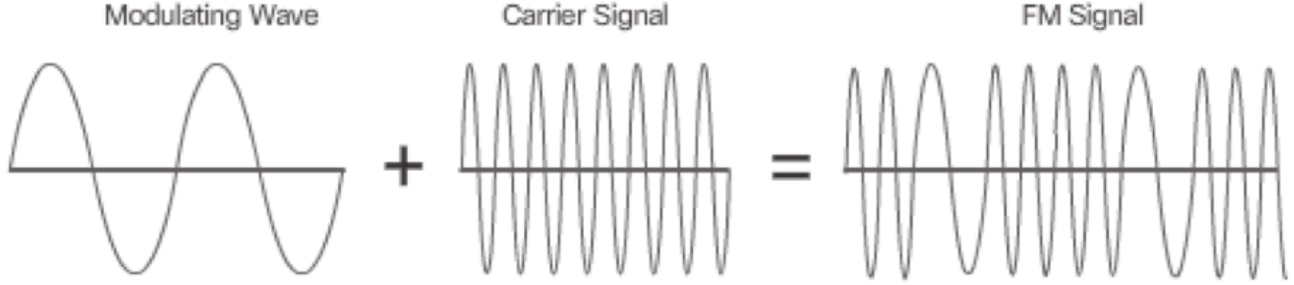
Media	Physical Components	Frame Encoding Technique	Signalling Method
Copper Cable	UTP Coaxial Connectors NICs Ports Interfaces	Manchester Encoding Non-Return to Zero (NRZ) techniques 4B/5B codes are used with Multi-Level Transition Level 3 (MLT-3) signaling 8B/10B PAM5	Changes in the electromagnetic field Intensity of the electromagnetic field Phase of the electromagnetic wave
Fiber Optic Cable	Single-mode Fiber Multimode Fiber Connectors NICs Interfaces Lasers and LEDs Photoreceptors	Pulses of light Wavelength multiplexing using different colors	A pulse equals 1. No pulse is 0.
Wireless Media	Access Points NICs Radio Antennae	DSSS (direct-sequence spread-spectrum) OFDM (orthogonal frequency division multiplexing)	Radio waves

Coderingstechniek – transmissie techniek

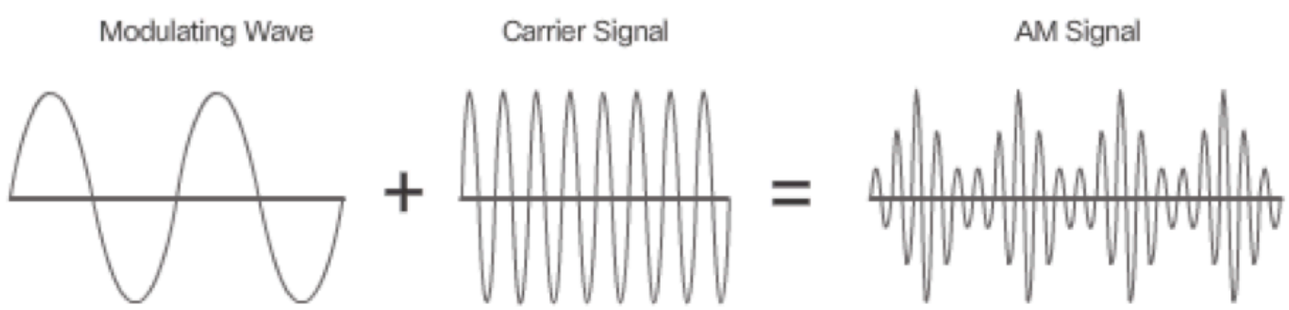
Manchester Encoding

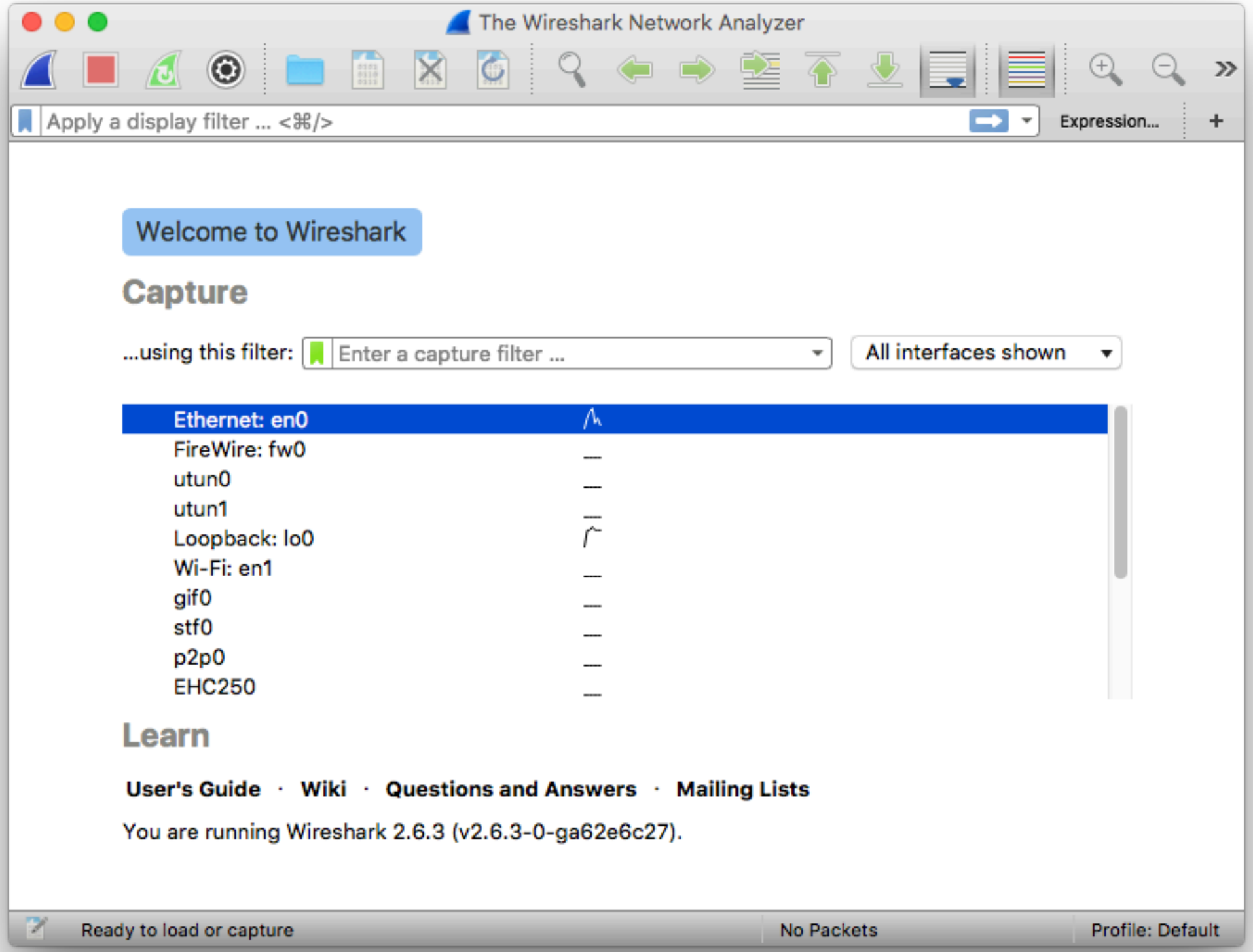


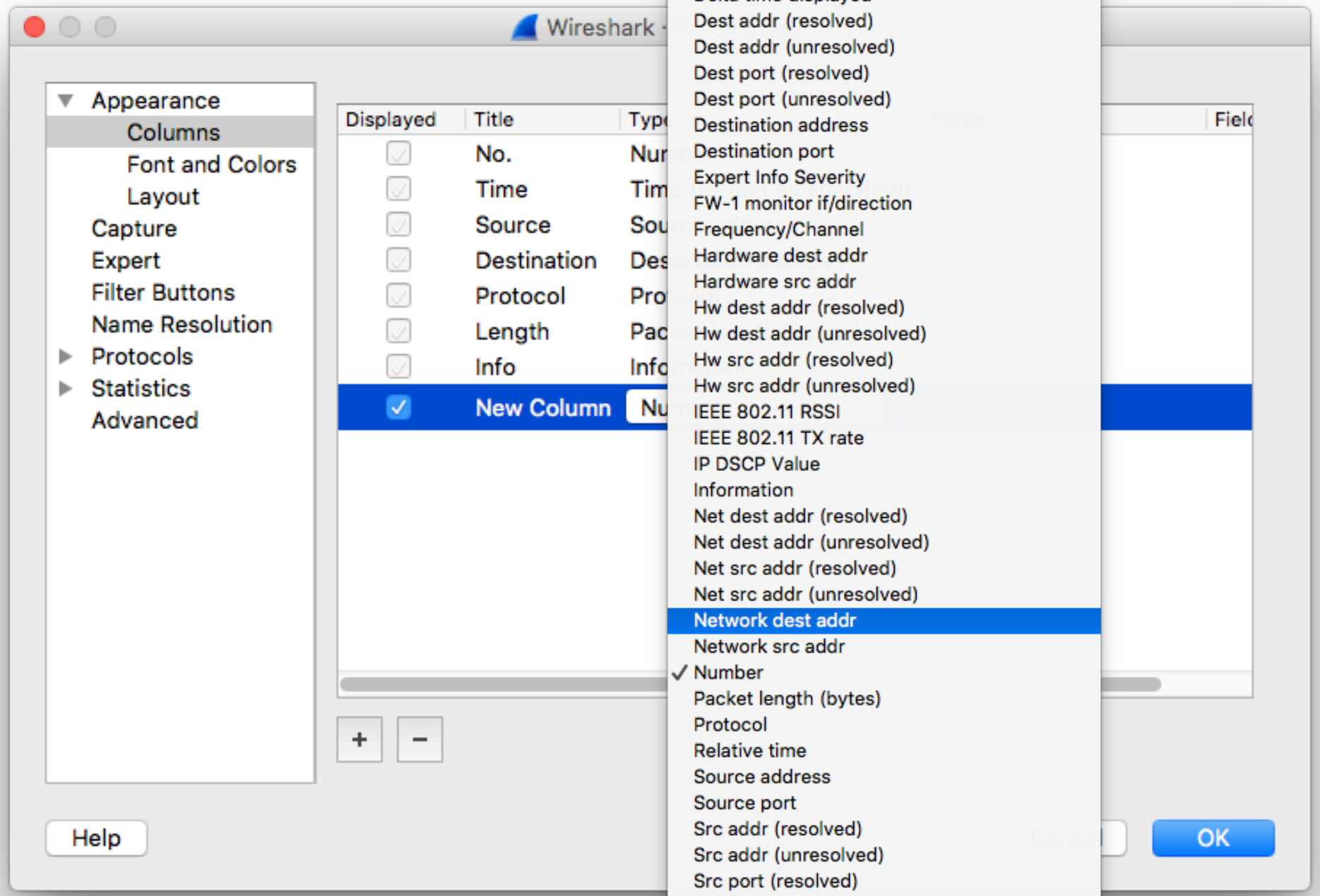
Frequency Modulation (FM)

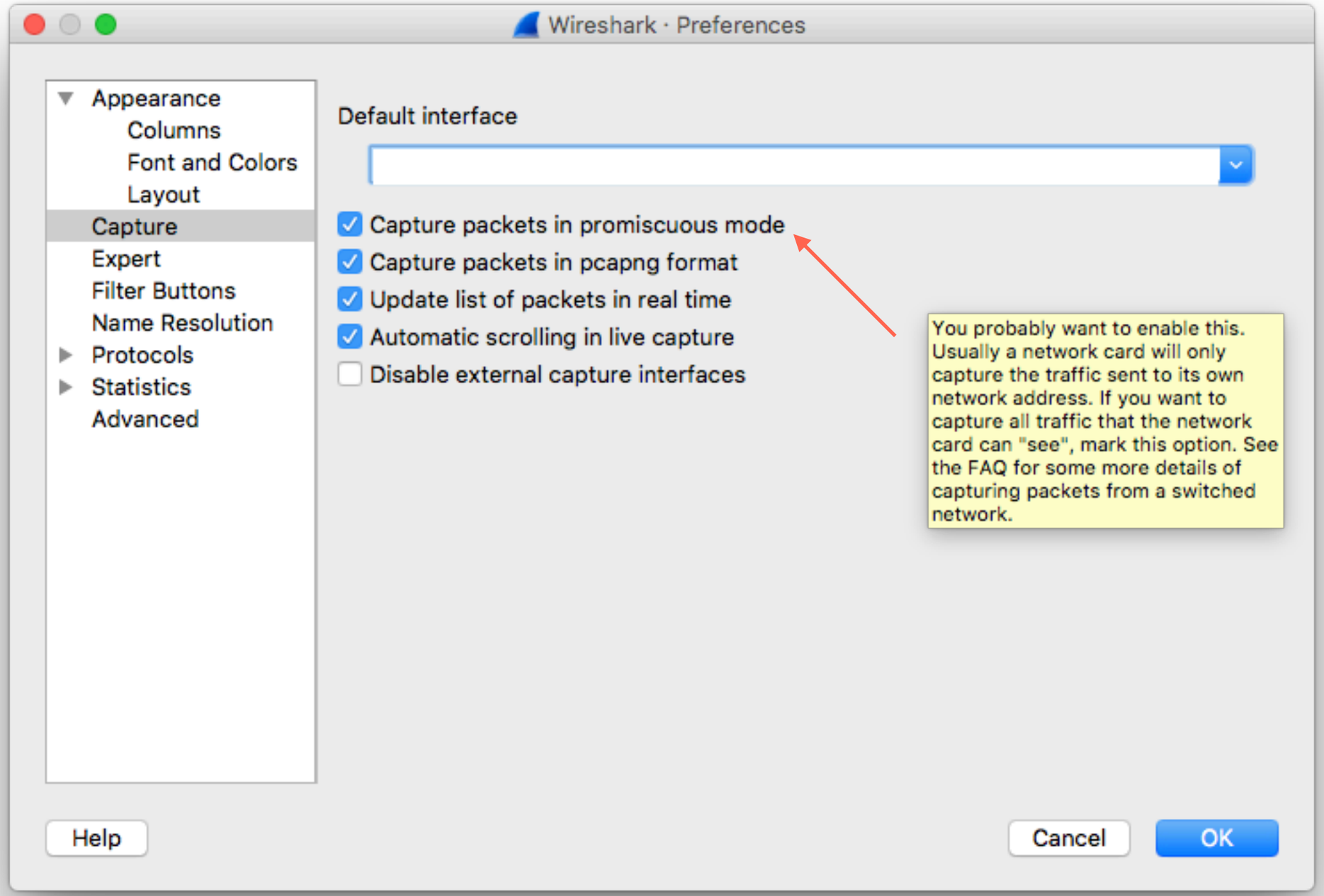


Amplitude Modulation (AM)




The Wireshark Network Analyzer
Apply a display filter ... <⌘/>
Expression...
+
Welcome to Wireshark
Capture
...using this filter:
Enter a capture filter ...
All interfaces shown
Ethernet: en0
FireWire: fw0
utun0
utun1
Loopback: lo0
Wi-Fi: en1
gif0
stf0
p2p0
EHC250
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists
You are running Wireshark 2.6.3 (v2.6.3-0-ga62e6c27).
Ready to load or capture
No Packets
Profile: Default


Wireshark
Appearance
Columns
Font and Colors
Layout
Capture
Expert
Filter Buttons
Name Resolution
Protocols
Statistics
Advanced
Displayed
Title
No.
Time
Source
Destination
Protocol
Length
Info
New Column
Help
OK
802.1Q VLAN id
Absolute date, as YYYY-MM-DD, and time
Absolute date, as YYYY/DOY, and time
Absolute time
Cisco VSAN
Cumulative Bytes
Custom
DCE/RPC call (cn_call_id / dg_seqnum)
Delta time
Delta time displayed
Dest addr (resolved)
Dest addr (unresolved)
Dest port (resolved)
Dest port (unresolved)
Destination address
Destination port
Expert Info Severity
FW-1 monitor if/direction
Frequency/Channel
Hardware dest addr
Hardware src addr
Hw dest addr (resolved)
Hw dest addr (unresolved)
Hw src addr (resolved)
Hw src addr (unresolved)
IEEE 802.11 RSSI
IEEE 802.11 TX rate
IP DSCP Value
Information
Net dest addr (resolved)
Net dest addr (unresolved)
Net src addr (resolved)
Net src addr (unresolved)
Network dest addr
Network src addr
Number
Packet length (bytes)
Protocol
Relative time
Source address
Source port
Src addr (resolved)
Src addr (unresolved)
Src port (resolved)
Src port (unresolved)
TEI
UTC date, as YYYY-MM-DD, and time
UTC date, as YYYY/DOY, and time
UTC time
Time (format as specified)


Wireshark · Preferences
Appearance
Columns
Font and Colors
Layout
Capture
Expert
Filter Buttons
Name Resolution
Protocols
Statistics
Advanced
Default interface
Capture packets in promiscuous mode
Capture packets in pcapng format
Update list of packets in real time
Automatic scrolling in live capture
Disable external capture interfaces
You probably want to enable this. Usually a network card will only capture the traffic sent to its own network address. If you want to capture all traffic that the network card can "see", mark this option. See the FAQ for some more details of capturing packets from a switched network.
Help
Cancel
OK

Ethernet: en0

Apply a display filter ... <⌘/> Expression... +

No.	Time	Source	Destination	Protocol	Length	Info
1022	25.708348	192.168.31.124	104.40.147.142	TCP	66	61220 → 443 [ACK] Seq=39 Ack=8591 Wi…
1023	25.730722	104.40.147.142	192.168.31.124	TLSv1…	247	Application Data
1024	25.730813	192.168.31.124	104.40.147.142	TCP	66	61220 → 443 [ACK] Seq=39 Ack=8772 Wi…
1025	25.733327	192.168.31.124	8.8.8.8	DNS	91	Standard query 0xd489 A lp-push-serv…
1026	25.755508	8.8.8.8	192.168.31.124	DNS	107	Standard query response 0xd489 A lp-…
1027	25.756064	192.168.31.124	192.241.181.178	TCP	78	63469 → 443 [SYN] Seq=0 Win=65535 Le…
1028	25.849826	192.241.181.178	192.168.31.124	TCP	74	443 → 63469 [SYN, ACK] Seq=0 Ack=1 W…
1029	25.849933	192.168.31.124	192.241.181.178	TCP	66	63469 → 443 [ACK] Seq=1 Ack=1 Win=13…
1030	25.850504	192.168.31.124	192.241.181.178	TLSv1	588	Client Hello
1031	25.999522	192.168.31.124	192.168.31.44	AFP	88	FPGetVolParms request: Vol=9
1032	25.999838	192.168.31.44	192.168.31.124	AFP	116	FPGetVolParms reply

▶ Frame 1026: 107 bytes on wire (856 bits), 107 bytes captured (856 bits) on interface 0
▶ Ethernet II, Src: BelkinIn_88:a2:20 (94:10:3e:88:a2:20), Dst: Apple_04:bc:28 (c4:2c:03:04:bc:28)
▶ Internet Protocol Version 4, Src: 8.8.8.8, Dst: 192.168.31.124
▶ User Datagram Protocol, Src Port: 53, Dst Port: 24806
▶ Domain Name System (response)

```
0000  c4 2c 03 04 bc 28 94 10  3e 88 a2 20 08 00 45 00   ·,···(·· >·· ··E·
0010  00 5d 81 6a 00 00 79 11  cf f1 08 08 08 08 c0 a8   ·]·j··y· ········
0020  1f 7c 00 35 60 e6 00 49  99 1c d4 89 81 80 00 01   ·|·5`··I ········
0030  00 01 00 00 00 00 12 6c  70 2d 70 75 73 68 2d 73   ·······l p-push-s
0040  65 72 76 65 72 2d 38 38  37 08 6c 61 73 74 70 61   erver-88 7·lastpa
0050  73 73 03 63 6f 6d 00 00  01 00 01 c0 0c 00 01 00   ss·com·· ········
0060  01 00 00 01 7e 00 04 c0  f1 b5 b2                  ····~··· ···
```

Ethernet (eth), 14 bytes Packets: 1066 · Displayed: 1066 (100.0%) · Dropped: 0 (0.0%) Profile: Default

Ethernet: en0

dns

Expression...

No.	Time	Source	Destination	Protocol	Length	Info
20	1.076600	192.168.31.124	8.8.8.8	DNS	88	Standard query 0x7ffd A 20.client-channel.…
21	1.097643	8.8.8.8	192.168.31.124	DNS	104	Standard query response 0x7ffd A 20.client…
954	24.587099	192.168.31.124	8.8.8.8	DNS	80	Standard query 0x8942 A adfarm.mediaplex.c…
958	24.664392	192.168.31.124	8.8.8.8	DNS	80	Standard query 0xbc9f A adfarm.mediaplex.c…
959	24.692671	8.8.8.8	192.168.31.124	DNS	255	Standard query response 0xbc9f A adfarm.me…
984	24.732391	8.8.8.8	192.168.31.124	DNS	255	Standard query response 0x8942 A adfarm.me…
1025	25.733327	192.168.31.124	8.8.8.8	DNS	91	Standard query 0xd489 A lp-push-server-887…
1026	25.755508	8.8.8.8	192.168.31.124	DNS	107	Standard query response 0xd489 A lp-push-s…

```
▶ Frame 1026: 107 bytes on wire (856 bits), 107 bytes captured (856 bits) on interface 0
▶ Ethernet II, Src: BelkinIn_88:a2:20 (94:10:3e:88:a2:20), Dst: Apple_04:bc:28 (c4:2c:03:04:bc:28)
▶ Internet Protocol Version 4, Src: 8.8.8.8, Dst: 192.168.31.124
▶ User Datagram Protocol, Src Port: 53, Dst Port: 24806
▶ Domain Name System (response)
```

```
0000  c4 2c 03 04 bc 28 94 10  3e 88 a2 20 08 00 45 00   ·,···(·· >·· ··E·
0010  00 5d 81 6a 00 00 79 11  cf f1 08 08 08 08 c0 a8   ·]·j··y· ········
0020  1f 7c 00 35 60 e6 00 49  99 1c d4 89 81 80 00 01   ·|·5`··I ········
0030  00 01 00 00 00 00 12 6c  70 2d 70 75 73 68 2d 73   ·······l p-push-s
0040  65 72 76 65 72 2d 38 38  37 08 6c 61 73 74 70 61   erver-88 7·lastpa
0050  73 73 03 63 6f 6d 00 00  01 00 01 c0 0c 00 01 00   ss·com·· ········
0060  01 00 00 01 7e 00 04 c0  f1 b5 b2                  ····~··· ···
```

Domain Name System: Protocol | Packets: 1066 · Displayed: 8 (0.8%) · Dropped: 0 (0.0%) | Profile: Default

Ethernet: en0

ip.src == 192.168.31.1 — Expression… +

No.	Time	Source	Destination	Protocol	Length	Info
136	4.810945	192.168.31.1	224.0.1.60	IGMPv2	60	Membership Query, specific for group 224.0…
137	4.811266	192.168.31.1	224.0.1.60	IGMPv2	60	Membership Query, specific for group 224.0…
138	4.811526	192.168.31.1	224.0.1.60	IGMPv2	60	Membership Query, specific for group 224.0…
454	9.981564	192.168.31.1	224.0.0.22	IGMPv2	60	Membership Query, specific for group 224.0…
455	9.982255	192.168.31.1	224.0.0.2	IGMPv2	60	Membership Query, specific for group 224.0…
483	10.636227	192.168.31.1	224.0.0.2	IGMPv2	60	Membership Report group 224.0.0.2
486	10.866153	192.168.31.1	224.0.0.22	IGMPv2	60	Membership Report group 224.0.0.22

```
▶ Frame 486: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
▶ Ethernet II, Src: BelkinIn_88:a2:20 (94:10:3e:88:a2:20), Dst: IPv4mcast_16 (01:00:5e:00:00:16)
▶ Internet Protocol Version 4, Src: 192.168.31.1, Dst: 224.0.0.22
▶ Internet Group Management Protocol
```

```
0000  01 00 5e 00 00 16 94 10  3e 88 a2 20 08 00 46 c0   ··^····· >·· ··F·
0010  00 20 00 00 40 00 01 02  24 58 c0 a8 1f 01 e0 00   · ··@··· $X······
0020  00 16 94 04 00 00 16 00  09 e9 e0 00 00 16 00 00   ········ ········
0030  00 00 00 00 00 00 00 00  77 2c 25 e8              ········ w,%·
```

Ethernet (eth), 28 bytes | Packets: 1066 · Displayed: 7 (0.7%) · Dropped: 0 (0.0%) | Profile: Default

Operators

Comparison operators:

Six comparison operators are available:

English format:	C like format:	Meaning:
eq	==	Equal
ne	!=	Not equal
gt	>	Greater than
lt	<	Less than
ge	>=	Greater or equal
le	<=	Less or equal

Logical expressions:

English format:	C like format:	Meaning:
and	&&	Logical AND
or	||	Logical OR
xor	^^	Logical XOR
not	!	Logical NOT

Filter Voorbeelden

Examples:

snmp || dns || icmp Display the SNMP or DNS or ICMP traffics.

ip.addr == 10.1.1.1

Displays the packets with source or destination IP address equals to 10.1.1.1.

ip.src != 10.1.2.3 or ip.dst != 10.4.5.6

Displays the packets with a source IP address different from 10.1.2.3 or with a destination IP different from 10.4.5.6.
In other words, the displayed packets will have:
Source IP address: anything but 10.1.2.3, destination IP address: anything
and
Source IP address: anything, destination IP address: anything but 10.4.5.6

ip.src != 10.1.2.3 and ip.dst != 10.4.5.6

Displays the packets with source IP different from 10.1.2.3 and in the same time with destination IP different from 10.4.5.6
In other words, the displayed packets will have:
Source IP address: anything but 10.1.2.3 and destination IP address: anything but 10.4.5.6

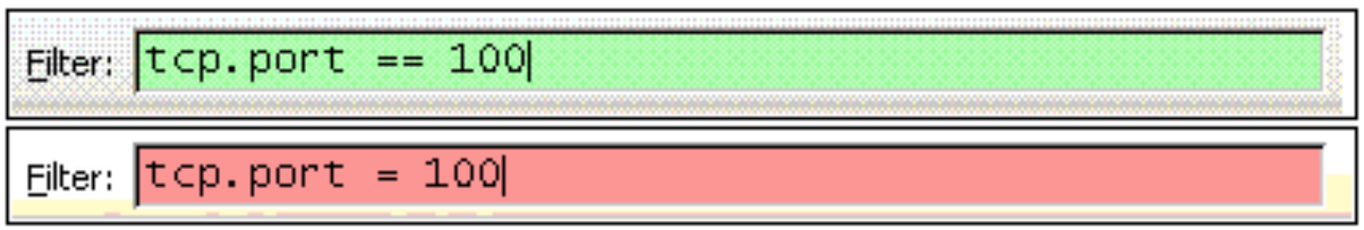
tcp.port == 25 Display packets with TCP source or destination port 25.

tcp.dstport == 25 Display packets with TCP destination port 25.

tcp.flags Display packets having a TCP flags

tcp.flags.syn == 0x02 Display packets with a TCP SYN flag.

If the filter syntax is correct, it will be highlighted in green, otherwise if there is a syntax mistake it will be highlighted in red.



https://www.wireshark.org/docs/wsug_html_chunked/ChWorkBuildDisplayFilterSection.html

Ethernet address

6 bytes separated by a colon (:), dot (.) or dash (-) with one or two bytes between separators:

```
eth.dst == ff:ff:ff:ff:ff:ff
eth.dst == ff-ff-ff-ff-ff-ff
eth.dst == ffff.ffff.ffff
```

IPv4 address

```
ip.addr == 192.168.0.1
```

Classless InterDomain Routing (CIDR) notation can be used to test if an IPv4 address is in a certain subnet. For example, this display filter will find all packets in the 129.111 Class-B network:

```
ip.addr == 129.111.0.0/16
```

IPv6 address

```
ipv6.addr == ::1
```

As with IPv4 addresses, IPv6 addresses can match a subnet.

Text string

```
http.request.uri == “https://www.wireshark.org/"
http.host matches "acme\.(orglcomlnet)"
```

6.4.4. Slice Operator

Wireshark allows you to select subsequences of a sequence in rather elaborate ways. After a label you can place a pair of brackets [] containing a comma separated list of range specifiers.

```
eth.src[0:3] == 00:00:83
```

The example above uses the n:m format to specify a single range. In this case n is the beginning offset and m is the length of the range being specified.

```
eth.src[1-2] == 00:83
```

The example above uses the n-m format to specify a single range. In this case n is the beginning offset and m is the ending offset.

```
eth.src[:4] == 00:00:83:00
```

The example above uses the :m format, which takes everything from the beginning of a sequence to offset m. It is equivalent to 0:m

```
eth.src[4:] == 20:20
```

The example above uses the n: format, which takes everything from offset n to the end of the sequence.

```
eth.src[2] == 83
```

The example above uses the n format to specify a single range. In this case the element in the sequence at offset n is selected. This is equivalent to n:1.

```
eth.src[0:3,1-2,:4,4:,2] ==
00:00:83:00:83:00:00:83:00:20:20:83
```

Wireshark allows you to string together single ranges in a comma separated list to form compound ranges as shown above.

6.4.5. Membership Operator

Wireshark allows you to test a field for membership in a set of values or fields. After the field name, use the in operator followed by the set items surrounded by braces {}.

```
tcp.port in {80 443 8080}
```

This can be considered a shortcut operator, as the previous expression could have been expressed as:

```
tcp.port == 80 || tcp.port == 443 || tcp.port == 8080
```

The set of values can also contain ranges:

```
tcp.port in {443 4430..4434}
```

This is not merely a shortcut for `tcp.port == 443 || (tcp.port >= 4430 && tcp.port <= 4434)`. Comparison operators are usually satisfied when any field matches the filter, and thus a packet with ports 80 and 56789 would match this alternative display filter since `56789 >= 4430 && 80 <= 4434` is true. The membership operator instead tests the same field against the range condition.

Sets are not just limited to numbers, other types can be used as well:

```
http.request.method in {"HEAD" "GET"}
ip.addr in {10.0.0.5 .. 10.0.0.9 192.168.1.1..192.168.1.9}
frame.time_delta in {10 .. 10.5}
```

6.4.7. A Common Mistake

Using the != operator on combined expressions like eth.addr, ip.addr, tcp.port, and udp.port will probably not work as expected. Wireshark will show the warning ""!=" is deprecated or may have unexpected results" when you use it.

Often people use a filter string to display something like `ip.addr == 1.2.3.4` which will display all packets containing the IP address 1.2.3.4.

Then they use `ip.addr != 1.2.3.4` to see all packets not containing the IP address 1.2.3.4 in it. Unfortunately, this does *not* do the expected.

Instead, that expression will even be true for packets where either source or destination IP address equals 1.2.3.4. The reason for this, is that the expression `ip.addr != 1.2.3.4` must be read as "the packet contains a field named ip.addr with a value different from 1.2.3.4". As an IP datagram contains both a source and a destination address, the expression will evaluate to true whenever at least one of the two addresses differs from 1.2.3.4.

If you want to filter out all packets containing IP datagrams to or from IP address 1.2.3.4, then the correct filter is `!(ip.addr == 1.2.3.4)` as it reads "show me all the packets for which it is not true that a field named ip.addr exists with a value of 1.2.3.4", or in other words, "filter out all packets for which there are no occurrences of a field named ip.addr with the value 1.2.3.4".

Extra voorbeelden filters

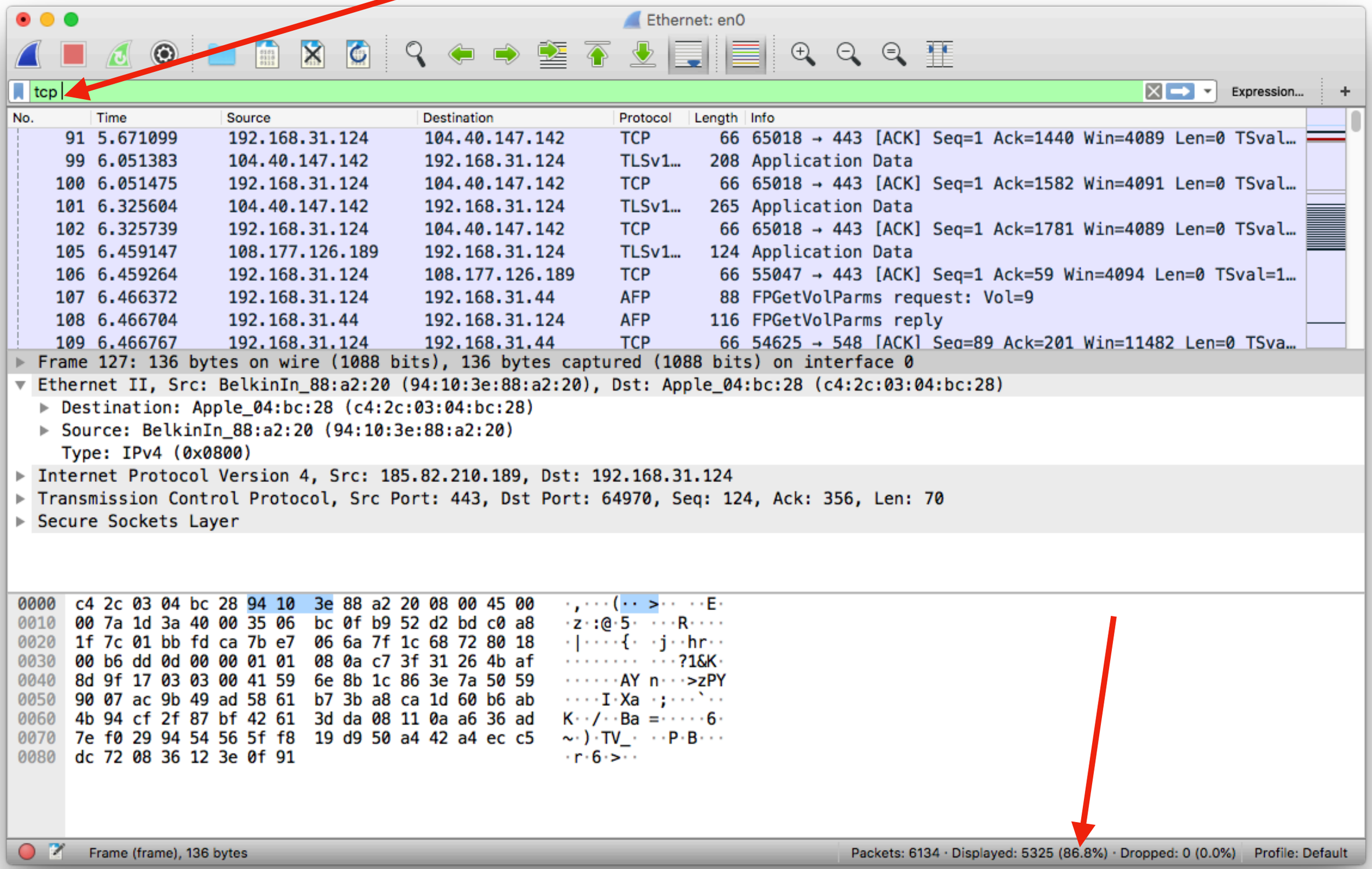
https://networksecuritytools.com/list-wireshark-display-filters/

```
▶ Frame 1: 97 bytes on wire (776 bits), 97 bytes captured (776 bits) on interface 0
▼ Ethernet II, Src: BelkinIn_88:a2:20 (94:10:3e:88:a2:20), Dst: Apple_04:bc:28 (c4:2c:03:04:bc:28)
  ▶ Destination: Apple_04:bc:28 (c4:2c:03:04:bc:28)
  ▶ Source: BelkinIn_88:a2:20 (94:10:3e:88:a2:20)
    Type: IPv4 (0x0800)
▶ Internet Protocol Version 4, Src: 217.148.93.160, Dst: 192.168.31.124
▶ Transmission Control Protocol, Src Port: 443, Dst Port: 59173, Seq: 1, Ack: 1, Len: 31
▶ Secure Sockets Layer
```

```
0000  c4 2c 03 04 bc 28 94 10  3e 88 a2 20 08 00 45 00   ·,···(·· >·· ··E·
0010  00 53 79 12 40 00 35 06  b5 39 d9 94 5d a0 c0 a8   ·Sy·@·5· ·9··]···
0020  1f 7c 01 bb e7 25 47 44  43 85 0b 5b 22 30 80 18   ·|···%GD C··["0··
0030  00 fd ad 96 00 00 01 01  08 0a c5 86 7e 9a 4b af   ········ ····~·K·
0040  12 9
0050  50 8
0060  89
```


```
▶ Frame 1: 97 bytes on wire (776 bits), 97 bytes captured (776 bits) on interface 0
▼ Ethernet II, Src: BelkinIn_88:a2:20 (94:10:3e:88:a2:20), Dst: Apple_04:bc:28 (c4:2c:03:04:bc:28)
  ▶ Destination: Apple_04:bc:28 (c4:2c:03:04:bc:28)
  ▶ Source: BelkinIn_88:a2:20 (94:10:3e:88:a2:20)
    Type: IPv4 (0x0800)
▶ Internet Protocol Version 4, Src: 217.148.93.160, Dst: 192.168.31.124
▶ Transmission Control Protocol, Src Port: 443, Dst Port: 59173, Seq: 1, Ack: 1, Len: 31
▶ Secure Sockets Layer
```

```
0000  c4 2c 03 04 bc 28 94 10  3e 88 a2 20 08 00 45 00   ·,···(·· >·· ··E·
0010  00 53 79 12 40 00 35 06  b5 39 d9 94 5d a0 c0 a8   ·Sy·@·5· ·9··]···
0020  1f 7c 01 bb e7 25 47 44  43 85 0b 5b 22 30 80 18   ·|···%GD C··["0··
0030  00 fd ad 96 00 00 01 01  08 0a c5 86 7e 9a 4b af   ········ ····~·K·
0040  12 91 17 03 03 00 1a fd  7d 2e a9 3e 14 4a 4d ec   ········ }.·>·JM·
0050  50 88 67 d4 be cf eb 0d  d2 2d 24 2c 0d dc bf fd   P·g····· ·-$,····
0060  89                                                ·
```


Ethernet: en0
tcp
Expression...
No. Time Source Destination Protocol Length Info
91 5.671099 192.168.31.124 104.40.147.142 TCP 66 65018 → 443 [ACK] Seq=1 Ack=1440 Win=4089 Len=0 TSval…
99 6.051383 104.40.147.142 192.168.31.124 TLSv1… 208 Application Data
100 6.051475 192.168.31.124 104.40.147.142 TCP 66 65018 → 443 [ACK] Seq=1 Ack=1582 Win=4091 Len=0 TSval…
101 6.325604 104.40.147.142 192.168.31.124 TLSv1… 265 Application Data
102 6.325739 192.168.31.124 104.40.147.142 TCP 66 65018 → 443 [ACK] Seq=1 Ack=1781 Win=4089 Len=0 TSval…
105 6.459147 108.177.126.189 192.168.31.124 TLSv1… 124 Application Data
106 6.459264 192.168.31.124 108.177.126.189 TCP 66 55047 → 443 [ACK] Seq=1 Ack=59 Win=4094 Len=0 TSval=1…
107 6.466372 192.168.31.124 192.168.31.44 AFP 88 FPGetVolParms request: Vol=9
108 6.466704 192.168.31.44 192.168.31.124 AFP 116 FPGetVolParms reply
109 6.466767 192.168.31.124 192.168.31.44 TCP 66 54625 → 548 [ACK] Seq=89 Ack=201 Win=11482 Len=0 TSva…
Frame 127: 136 bytes on wire (1088 bits), 136 bytes captured (1088 bits) on interface 0
Ethernet II, Src: BelkinIn_88:a2:20 (94:10:3e:88:a2:20), Dst: Apple_04:bc:28 (c4:2c:03:04:bc:28)
Destination: Apple_04:bc:28 (c4:2c:03:04:bc:28)
Source: BelkinIn_88:a2:20 (94:10:3e:88:a2:20)
Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 185.82.210.189, Dst: 192.168.31.124
Transmission Control Protocol, Src Port: 443, Dst Port: 64970, Seq: 124, Ack: 356, Len: 70
Secure Sockets Layer
0000 c4 2c 03 04 bc 28 94 10 3e 88 a2 20 08 00 45 00
0010 00 7a 1d 3a 40 00 35 06 bc 0f b9 52 d2 bd c0 a8
0020 1f 7c 01 bb fd ca 7b e7 06 6a 7f 1c 68 72 80 18
0030 00 b6 dd 0d 00 00 01 01 08 0a c7 3f 31 26 4b af
0040 8d 9f 17 03 03 00 41 59 6e 8b 1c 86 3e 7a 50 59
0050 90 07 ac 9b 49 ad 58 61 b7 3b a8 ca 1d 60 b6 ab
0060 4b 94 cf 2f 87 bf 42 61 3d da 08 11 0a a6 36 ad
0070 7e f0 29 94 54 56 5f f8 19 d9 50 a4 42 a4 ec c5
0080 dc 72 08 36 12 3e 0f 91
Frame (frame), 136 bytes
Packets: 6134 · Displayed: 5325 (86.8%) · Dropped: 0 (0.0%)
Profile: Default

ICMP = ping

Ethernet: en0

icmp

Expression… +

No.	Time	Source	Destination	Protocol	Length	Info
184	7.116482	192.168.31.124	8.8.8.8	ICMP	70	Destination unreachable (Port unreachable)
384	8.757787	192.168.31.124	172.217.17.110	ICMP	98	Echo (ping) request id=0xa8b2, seq=0/0, ttl=64 (reply in …
385	8.787625	172.217.17.110	192.168.31.124	ICMP	98	Echo (ping) reply id=0xa8b2, seq=0/0, ttl=51 (request i…
539	9.761021	192.168.31.124	172.217.17.110	ICMP	98	Echo (ping) request id=0xa8b2, seq=1/256, ttl=64 (reply i…
540	9.787157	172.217.17.110	192.168.31.124	ICMP	98	Echo (ping) reply id=0xa8b2, seq=1/256, ttl=51 (request…
589	10.764882	192.168.31.124	172.217.17.110	ICMP	98	Echo (ping) request id=0xa8b2, seq=2/512, ttl=64 (reply i…
590	10.796414	172.217.17.110	192.168.31.124	ICMP	98	Echo (ping) reply id=0xa8b2, seq=2/512, ttl=51 (request…
4590	44.831304	192.168.31.1	192.168.31.124	ICMP	130	Destination unreachable (Port unreachable)

```
▶ Frame 385: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
▼ Ethernet II, Src: BelkinIn_88:a2:20 (94:10:3e:88:a2:20), Dst: Apple_04:bc:28 (c4:2c:03:04:bc:28)
    ▶ Destination: Apple_04:bc:28 (c4:2c:03:04:bc:28)
    ▶ Source: BelkinIn_88:a2:20 (94:10:3e:88:a2:20)
      Type: IPv4 (0x0800)
▶ Internet Protocol Version 4, Src: 172.217.17.110, Dst: 19
▶ Internet Control Message Protocol
```

```
0000  c4 2c 03 04 bc 28 94 10  3e 88 a2 20 08 00 45 00   ·,···(·· >·· ··E·
0010  00 54 00 00 00 00 33 01  e9 3d ac d9 11 6e c0 a8   ·T····3· ·=···n··
0020  1f 7c 00 00 9c 0b a8 b2  00 00 5b b5 0d ee 00 01   ·|···· ·· [·····
0030  66 9a 08 09 0a 0b 0c 0d  0e 0f 10 11 12 13 14 15   f······ ·· ·····
0040  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24 25   ····· ·· !"#$%
0050  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34 35   &'()*+,- ./012345
0060  36 37                                              67
```

iMac — -bash — 80×41

```
iMac27:~ iMac$ ping google.com
PING google.com (172.217.17.110): 56 data bytes
64 bytes from 172.217.17.110: icmp_seq=0 ttl=51 time=31.319 ms
64 bytes from 172.217.17.110: icmp_seq=1 ttl=51 time=27.952 ms
64 bytes from 172.217.17.110: icmp_seq=2 ttl=51 time=30.248 ms
64 bytes from 172.217.17.110: icmp_seq=3 ttl=51 time=33.228 ms
^C
--- google.com ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 27.952/30.687/33.228/1.906 ms
iMac27:~ iMac$
```

Frame (frame), 98 bytes — Packets: 5393 · Displayed: 8 (0.1%) · Dropped: 0 (0.0%) — Profile: Default

Ethernet: en0

udp.port == 53

Expression...

No.	Time	Source	Destination	Protocol	Length	Info
14	0.136677	192.168.31.124	8.8.8.8	DNS	83	Standard query 0x13be A col.eum-appdynamics.com
15	0.169169	8.8.8.8	192.168.31.124	DNS	211	Standard query response 0x13be A col.eum-appdynamics…
174	7.094961	192.168.31.124	8.8.8.8	DNS	74	Standard query 0xeaf8 A www.google.com
183	7.116396	8.8.8.8	192.168.31.124	DNS	90	Standard query response 0xeaf8 A www.google.com A 17…
184	7.116482	192.168.31.124	8.8.8.8	ICMP	70	Destination unreachable (Port unreachable)
256	7.386716	192.168.31.124	8.8.8.8	DNS	80	Standard query 0xaa8e A adfarm.mediaplex.com
269	7.432291	8.8.8.8	192.168.31.124	DNS	255	Standard query response 0xaa8e A adfarm.mediaplex.co…
382	8.735099	192.168.31.124	8.8.8.8	DNS	70	Standard query 0x5e95 A google.com
383	8.757068	8.8.8.8	192.168.31.124	DNS	86	Standard query response 0x5e95 A google.com A 172.21…
628	11.119230	192.168.31.124	8.8.8.8	DNS	81	Standard query 0xd8eb A outlook.office365.com

```
▶ Frame 383: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
▼ Ethernet II, Src: BelkinIn_88:a2:20 (94:10:3e:88:a2:20), Dst: Apple_04:bc:28 (c4:2c:03:04:bc:28)
  ▶ Destination: Apple_04:bc:28 (c4:2c:03:04:bc:28)
  ▶ Source: BelkinIn_88:a2:20 (94:10:3e:88:a2:20)
    Type: IPv4 (0x0800)
▶ Internet Protocol Version 4, Src: 8.8.8.8, Dst: 192.168.31.124
▶ User Datagram Protocol, Src Port: 53, Dst Port: 63382
▶ Domain Name System (response)
```

```
0000  c4 2c 03 04 bc 28 94 10  3e 88 a2 20 08 00 45 00   ·,···(·· >·· ··E·
0010  00 48 36 94 00 00 79 11  1a dd 08 08 08 08 c0 a8   ·H6···y· ········
0020  1f 7c 00 35 f7 96 00 34  a5 30 5e 95 81 80 00 01   ·|·5···4 ·0^·····
0030  00 01 00 00 00 00 06 67  6f 6f 67 6c 65 03 63 6f   ·······g oogle·co
0040  6d 00 00 01 00 01 c0 0c  00 01 00 01 00 00 01 2b   m·······  ·······+
0050  00 04 ac d9 11 6e                                 ·····n
```

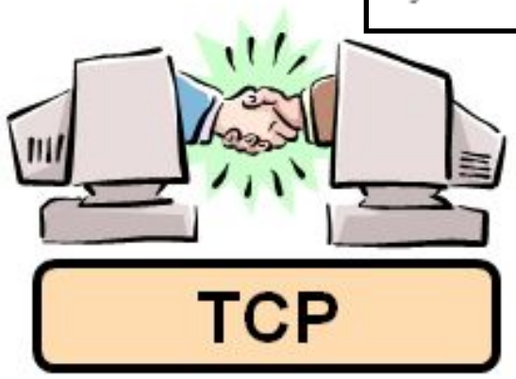
Frame (frame), 86 bytes

Packets: 5393 · Displayed: 29 (0.5%) · Dropped: 0 (0.0%)

Profile: Default

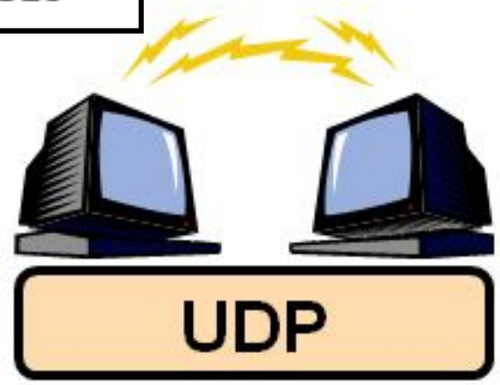
TCP VS UDP

Application	WWW	FTP	E-mail	NFS	VoIP	DNS
Transport	TCP			UDP		
Network	IP					
Physical	Ethernet		AAL-5	HDLC		



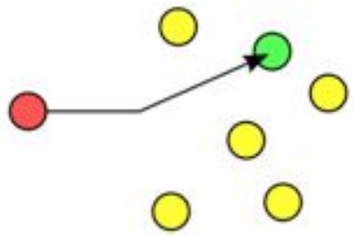
TCP

- **Slower but reliable transfers**
- **Typical applications:**
  - Email
  - Web browsing

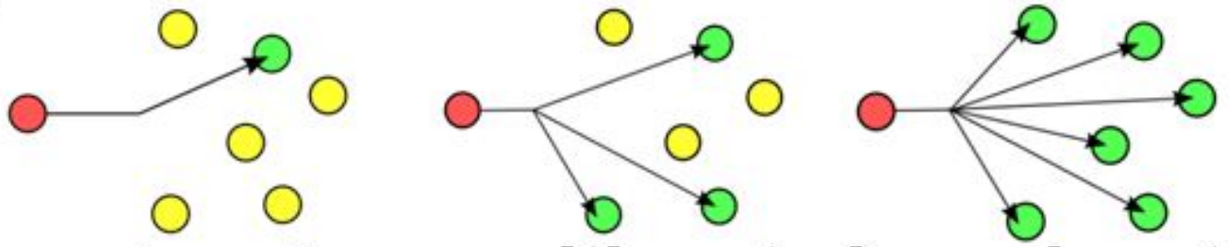


UDP

- **Fast but non-guaranteed transfers (“best effort”)**
- **Typical applications:**
  - VoIP
  - Music streaming

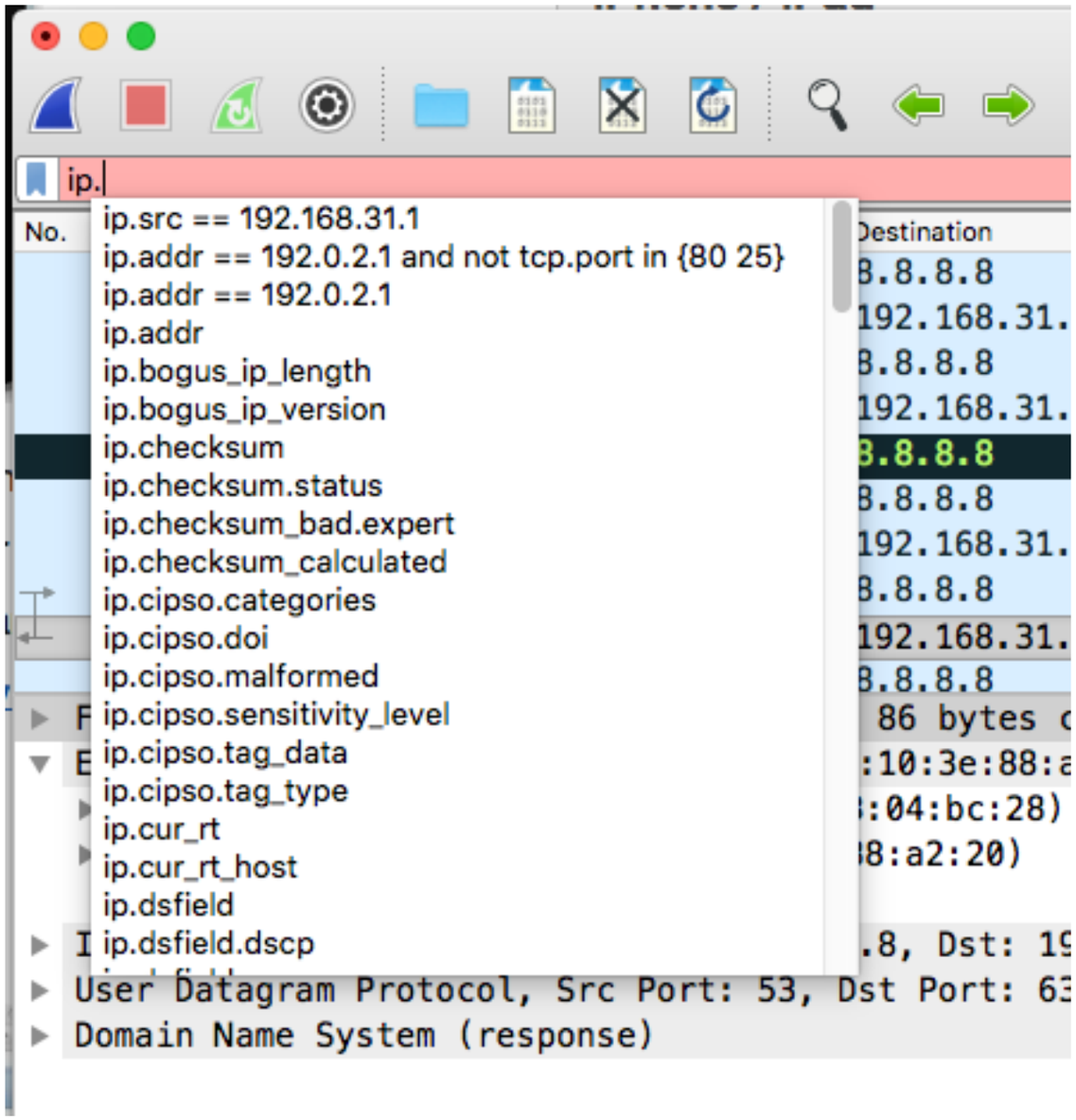


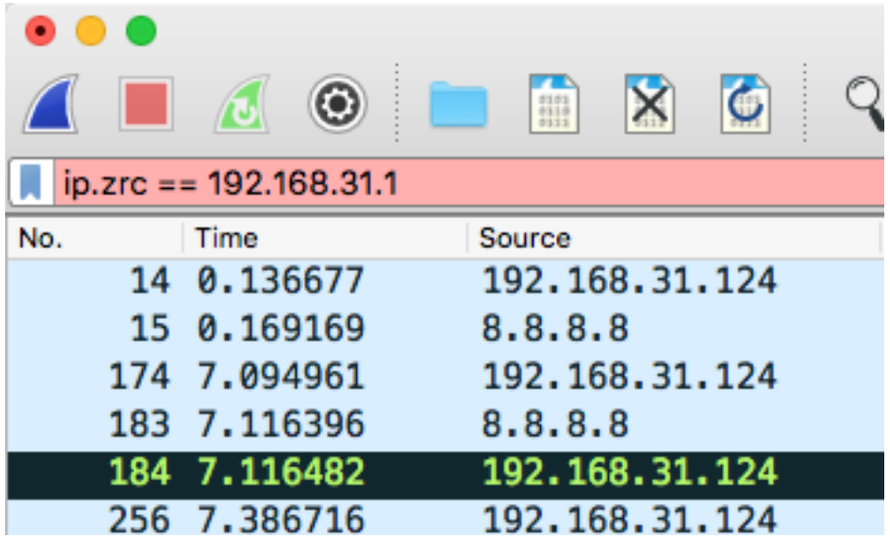
unicast

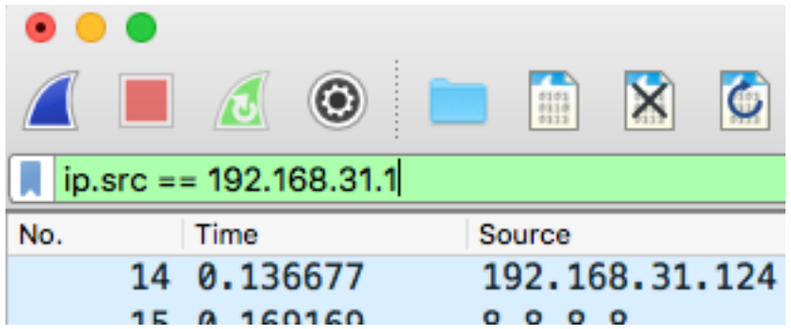


unicast **multicast** **broadcast**

Help


ip.
ip.src == 192.168.31.1
ip.addr == 192.0.2.1 and not tcp.port in {80 25}
ip.addr == 192.0.2.1
ip.addr
ip.bogus_ip_length
ip.bogus_ip_version
ip.checksum
ip.checksum.status
ip.checksum_bad.expert
ip.checksum_calculated
ip.cipso.categories
ip.cipso.doi
ip.cipso.malformed
ip.cipso.sensitivity_level
ip.cipso.tag_data
ip.cipso.tag_type
ip.cur_rt
ip.cur_rt_host
ip.dsfield
ip.dsfield.dscp
No.
Destination
8.8.8.8
192.168.31.
8.8.8.8
192.168.31.
8.8.8.8
8.8.8.8
192.168.31.
8.8.8.8
192.168.31.
86 bytes
:10:3e:88:a
:04:bc:28)
8:a2:20)
.8, Dst: 19
User Datagram Protocol, Src Port: 53, Dst Port: 63
Domain Name System (response)


ip.zrc == 192.168.31.1
No. Time Source
14 0.136677 192.168.31.124
15 0.169169 8.8.8.8
174 7.094961 192.168.31.124
183 7.116396 8.8.8.8
184 7.116482 192.168.31.124
256 7.386716 192.168.31.124

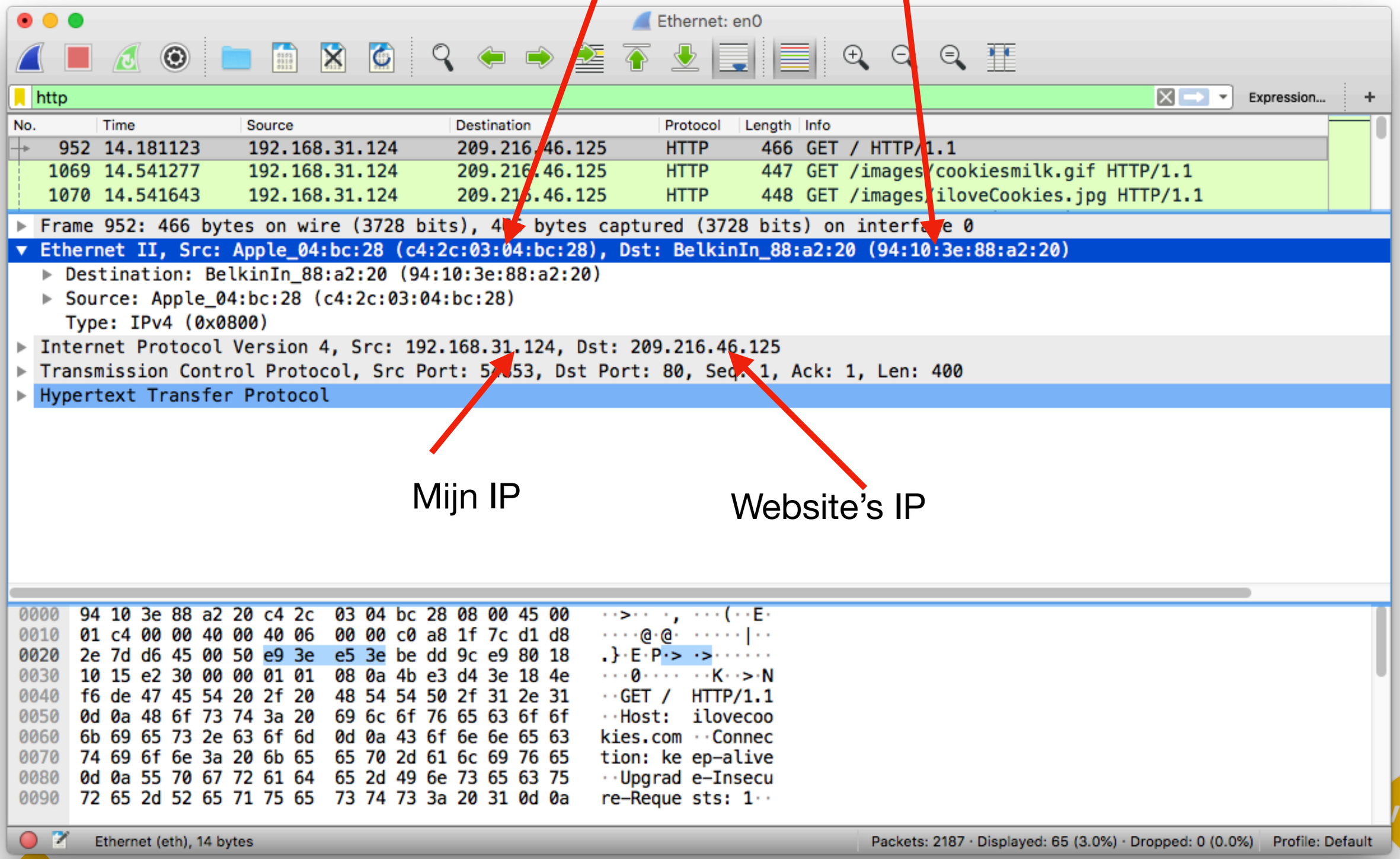

ip.src == 192.168.31.1
No. Time Source
14 0.136677 192.168.31.124

Packet Analysis

Frame = Physical layer

Mijn MAC

Router's MAC



Mijn IP

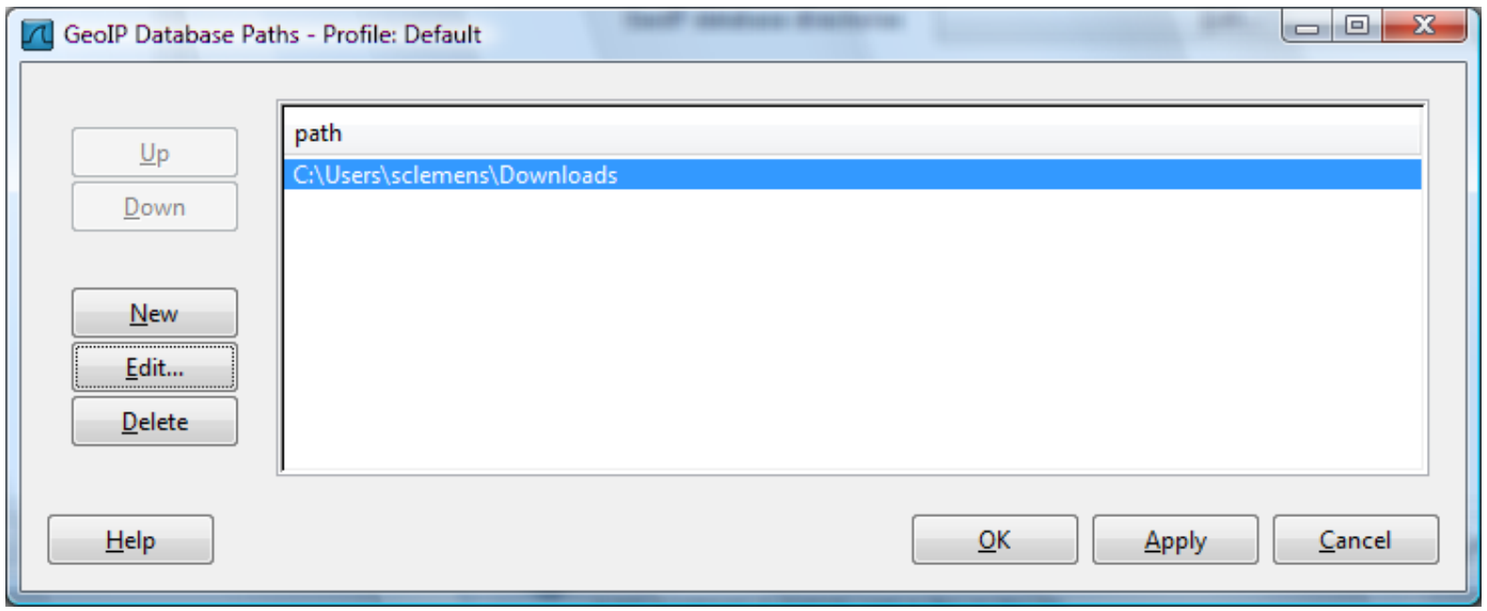
Website's IP

https://wiki.wireshark.org/HowToUseGeoIP

Wireshark does not ship with any GeoIP or GeoLite databases, so you have to download them yourself. You can get them at the following locations:

- GeoLite City, Country, and ASNum: http://geolite.maxmind.com/download/geoip/database/ (free download)
- GeoIP products: http://www.maxmind.com/app/products (purchase required)

It's more convenient if you put all of the databases in the same directory. Once you've downloaded your databases, you must tell Wireshark where they are. Go to *Edit→Preferences→Name Resolution* and select *GeoIP database directories*. Add the full path of each database directory, as shown below:



Filtering Traffic

You can use the *ip.geoip* display filters to filter traffic.

Exclude U.S.-based traffic:

```
ip and not ip.geoip.country == "United States"
```

Show address above the arctic circle:

```
ip.geoip.lat > "66.5"
```

Ethernet: en0

http

Expression...

No.	Time	Source	Destination	Protocol	Length	Info
428	5.411763	192.168.31.124	145.239.69.56	HTTP	462	GET / HTTP/1.1
432	5.434108	145.239.69.56	192.168.31.124	HTTP	703	HTTP/1.1 301 Moved Permanently (text/html)
1568	6.398795	192.168.31.124	2.22.55.123	HTTP	341	GET /MFYwVKADAgEAME0wSzBJMAkGBSsOAwIaBQAEF…

```
▶ Frame 428: 462 bytes on wire (3696 bits), 462 bytes captured (3696 bits) on interface 0
▶ Ethernet II, Src: Apple_04:bc:28 (c4:2c:03:04:bc:28), Dst: BelkinIn_88:a2:20 (94:10:3e:88:a2:20)
▶ Internet Protocol Version 4, Src: 192.168.31.124, Dst: 145.239.69.56
▶ Transmission Control Protocol, Src Port: 59094, Dst Port: 80, Seq: 1, Ack: 1, Len: 396
▼ Hypertext Transfer Protocol
  ▶ GET / HTTP/1.1\r\n
    Host: www.miras.be\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.100 Safa
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-US,en;q=0.9,nl;q=0.8\r\n
    \r\n
    [Full request URI: http://www.miras.be/]
    [HTTP request 1/1]
    [Response in frame: 432]
```

```
0020  45 38 e6 d6 00 50 0d 00  01 ca 65 0a 00 8e 80 18   E8···P·· ··e·····
0030  10 15 b8 fe 00 00 01 01  08 0a 4b f1 db ae 18 2d   ········ ··K····-
0040  8e f0 47 45 54 20 2f 20  48 54 54 50 2f 31 2e 31   ··GET /  HTTP/1.1
0050  0d 0a 48 6f 73 74 3a 20  77 77 77 2e 6d 69 72 61   ··Host:  www.mira
0060  73 2e 62 65 0d 0a 43 6f  6e 6e 65 63 74 69 6f 6e   s.be··Co nnection
0070  3a 20 6b 65 65 70 2d 61  6c 69 76 65 0d 0a 55 70   : keep-a live··Up
0080  67 72 61 64 65 2d 49 6e  73 65 63 75 72 65 2d 52   grade-In secure-R
0090  65 71 75 65 73 74 73 3a  20 31 0d 0a 55 73 65 72   equests:  1··User
00a0  2d 41 67 65 6e 74 3a 20  4d 6f 7a 69 6c 6c 61 2f   -Agent:  Mozilla/
00b0  35 2e 30 20 28 4d 61 63  69 6e 74 6f 73 68 3b 20   5.0 (Mac intosh;
00c0  49 6e 74 65 6c 20 4d 61  63 20 4f 53 20 58 20 31   Intel Ma c OS X 1
```

Transmission Control Protocol (tcp), 32 bytes

Packets: 11171 · Displayed: 4 (0.0%) · Dropped: 0 (0.0%)

Profile: Default

HTTP Request

Ethernet: en0

http

Expression...

No.	Time	Source	Destination	Protocol	Length	Info
428	5.411763	192.168.31.124	145.239.69.56	HTTP	462	GET / HTTP/1.1
432	5.434108	145.239.69.56	192.168.31.124	HTTP	703	HTTP/1.1 301 Moved Permanently (text/html)
1568	6.398795	192.168.31.124	2.22.55.123	HTTP	341	GET /MFYwVKADAgEAME0wSzBJMAkGBSsOAwIaBQAEF…

```
▶ Frame 428: 462 bytes on wire (3696 bits), 462 bytes captured (3696 bits) on interface 0
▶ Ethernet II, Src: Apple_04:bc:28 (c4:2c:03:04:bc:28), Dst: BelkinIn_88:a2:20 (94:10:3e:88:a2:20)
▶ Internet Protocol Version 4, Src: 192.168.31.124, Dst: 145.239.69.56
▶ Transmission Control Protocol, Src Port: 59094, Dst Port: 80, Seq: 1, Ack: 1, Len: 396
▼ Hypertext Transfer Protocol
  ▶ GET / HTTP/1.1\r\n
    Host: www.miras.be\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.100 Safa
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-US,en;q=0.9,nl;q=0.8\r\n
    \r\n
    [Full request URI: http://www.miras.be/]
    [HTTP request 1/1]
    [Response in frame: 432]
```

```
0020  45 38 e6 d6 00 50 0d 00  01 ca 65 0a 00 8e 80 18   E8···P·· ··e·····
0030  10 15 b8 fe 00 00 01 01  08 0a 4b f1 db ae 18 2d   ········ ··K····-
0040  8e f0 47 45 54 20 2f 20  48 54 54 50 2f 31 2e 31   ··GET /  HTTP/1.1
0050  0d 0a 48 6f 73 74 3a 20  77 77 77 2e 6d 69 72 61   ··Host:  www.mira
0060  73 2e 62 65 0d 0a 43 6f  6e 6e 65 63 74 69 6f 6e   s.be··Co nnection
0070  3a 20 6b 65 65 70 2d 61  6c 69 76 65 0d 0a 55 70   : keep-a live··Up
0080  67 72 61 64 65 2d 49 6e  73 65 63 75 72 65 2d 52   grade-In secure-R
0090  65 71 75 65 73 74 73 3a  20 31 0d 0a 55 73 65 72   equests:  1··User
00a0  2d 41 67 65 6e 74 3a 20  4d 6f 7a 69 6c 6c 61 2f   -Agent:  Mozilla/
00b0  35 2e 30 20 28 4d 61 63  69 6e 74 6f 73 68 3b 20   5.0 (Mac intosh;
00c0  49 6e 74 65 6c 20 4d 61  63 20 4f 53 20 58 20 31   Intel Ma c OS X 1
```

Transmission Control Protocol (tcp), 32 bytes

Packets: 11171 · Displayed: 4 (0.0%) · Dropped: 0 (0.0%)

Profile: Default

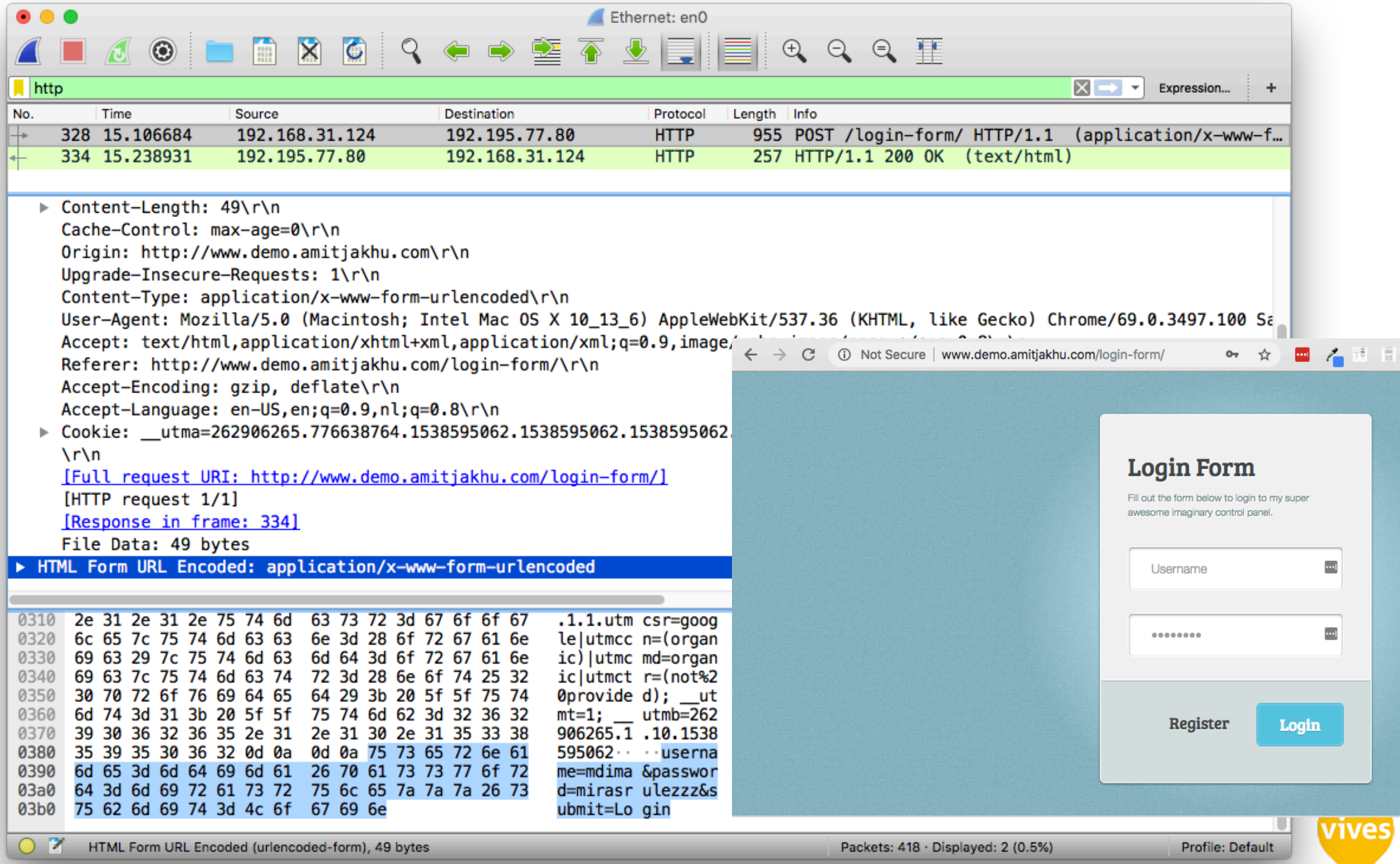
About Chrome

Google Chrome

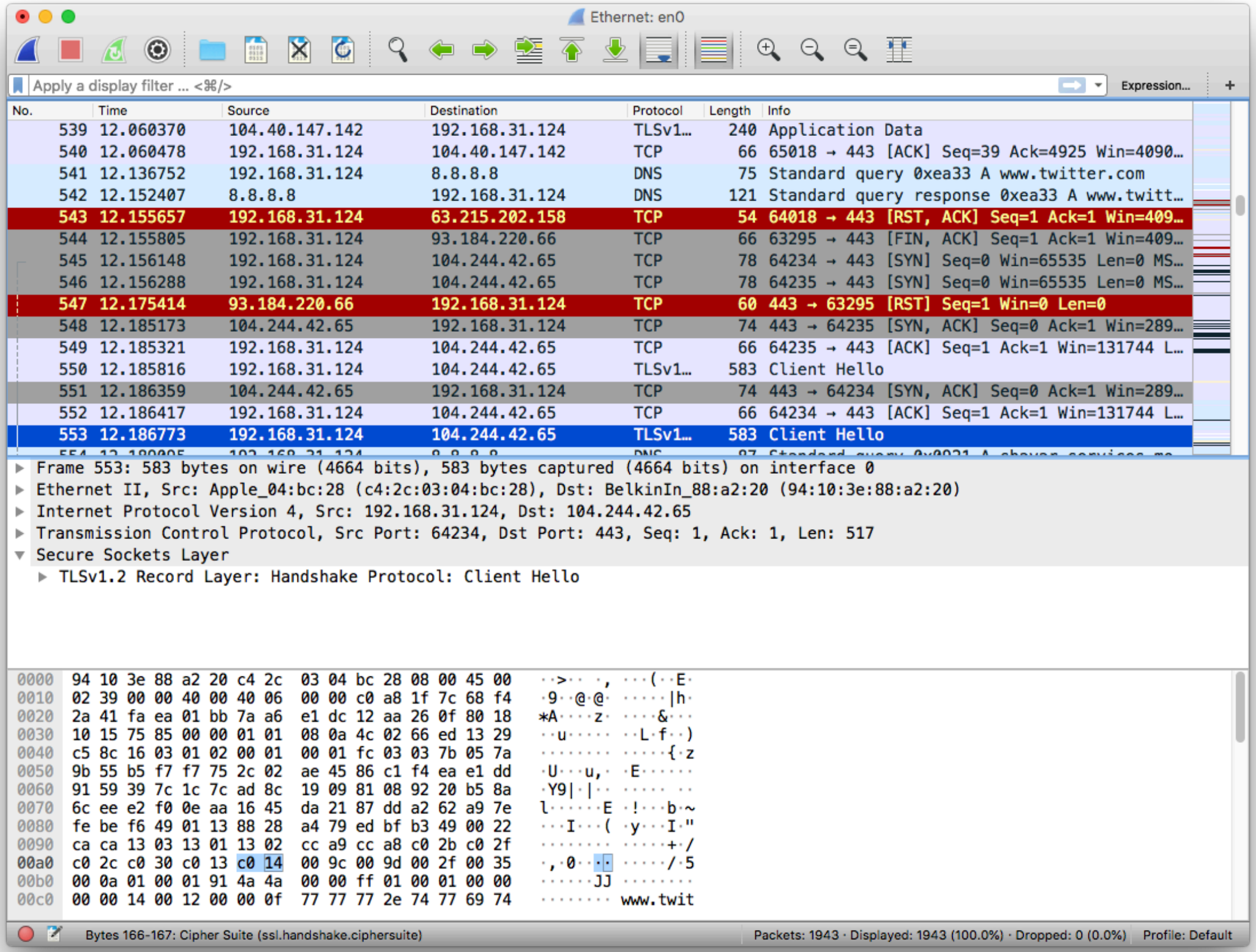
Google Chrome is up to date

Version 69.0.3497.100 (Official Build) (64-bit)

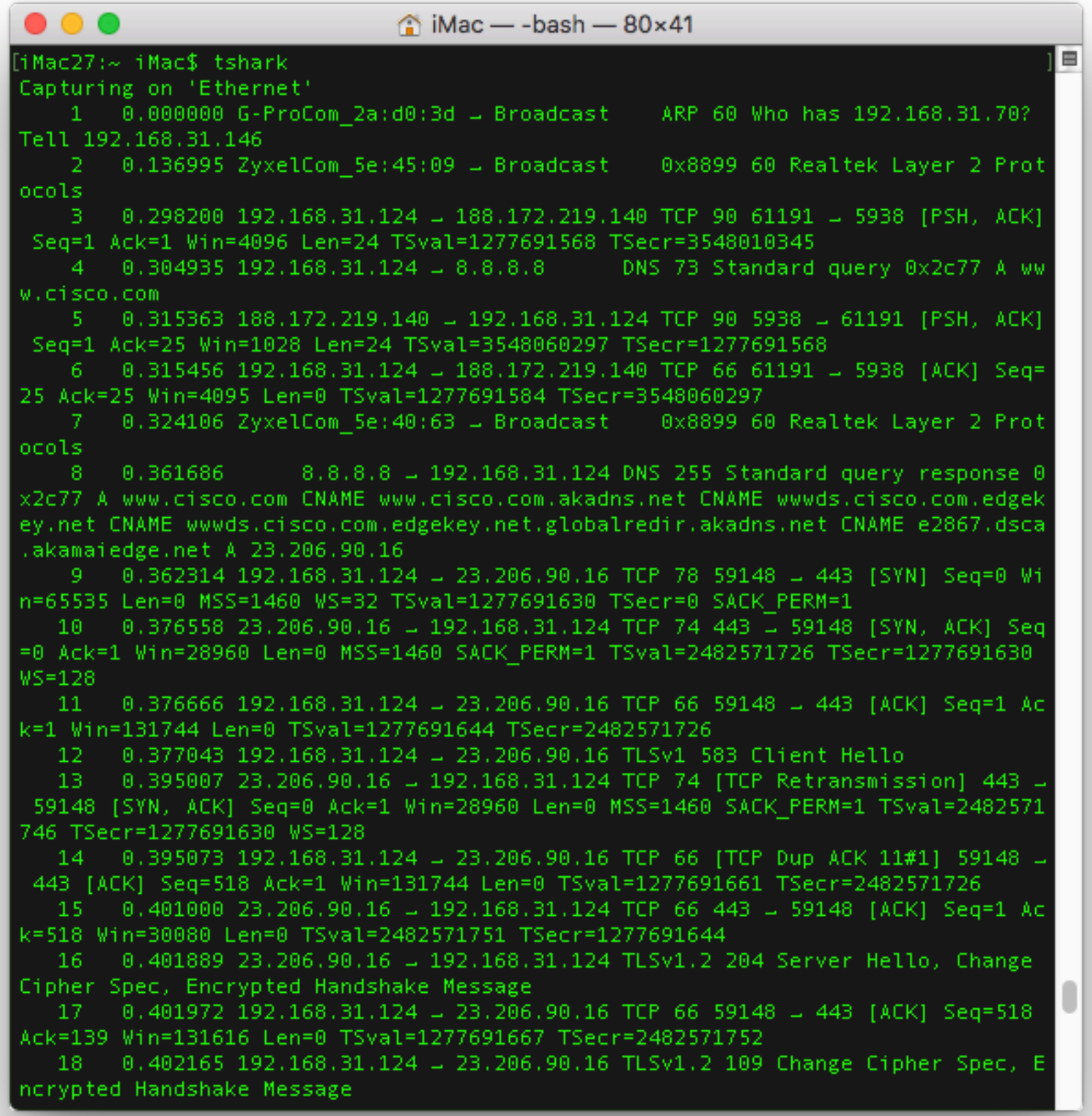
HTTP


Ethernet: en0
http
Expression...
No. Time Source Destination Protocol Length Info
328 15.106684 192.168.31.124 192.195.77.80 HTTP 955 POST /login-form/ HTTP/1.1 (application/x-www-f…
334 15.238931 192.195.77.80 192.168.31.124 HTTP 257 HTTP/1.1 200 OK (text/html)
Content-Length: 49\r\n
Cache-Control: max-age=0\r\n
Origin: http://www.demo.amitjakhu.com\r\n
Upgrade-Insecure-Requests: 1\r\n
Content-Type: application/x-www-form-urlencoded\r\n
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.100 Sa
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/
Referer: http://www.demo.amitjakhu.com/login-form/\r\n
Accept-Encoding: gzip, deflate\r\n
Accept-Language: en-US,en;q=0.9,nl;q=0.8\r\n
Cookie: __utma=262906265.776638764.1538595062.1538595062.1538595062.
\r\n
[Full request URI: http://www.demo.amitjakhu.com/login-form/]
[HTTP request 1/1]
[Response in frame: 334]
File Data: 49 bytes
HTML Form URL Encoded: application/x-www-form-urlencoded
0310 2e 31 2e 31 2e 75 74 6d 63 73 72 3d 67 6f 6f 67 .1.1.utm csr=goog
0320 6c 65 7c 75 74 6d 63 63 6e 3d 28 6f 72 67 61 6e le|utmcc n=(organ
0330 69 63 29 7c 75 74 6d 63 6d 64 3d 6f 72 67 61 6e ic)|utmc md=organ
0340 69 63 7c 75 74 6d 63 74 72 3d 28 6e 6f 74 25 32 ic|utmct r=(not%2
0350 30 70 72 6f 76 69 64 65 64 29 3b 20 5f 5f 75 74 0provide d); __ut
0360 6d 74 3d 31 3b 20 5f 5f 75 74 6d 62 3d 32 36 32 mt=1; __ utmb=262
0370 39 30 36 32 36 35 2e 31 2e 31 30 2e 31 35 33 38 906265.1 .10.1538
0380 35 39 35 30 36 32 0d 0a 0d 0a 75 73 65 72 6e 61 595062·· ··userna
0390 6d 65 3d 6d 64 69 6d 61 26 70 61 73 73 77 6f 72 me=mdima &passwor
03a0 64 3d 6d 69 72 61 73 72 75 6c 65 7a 7a 7a 26 73 d=mirasr ulezzz&s
03b0 75 62 6d 69 74 3d 4c 6f 67 69 6e ubmit=Lo gin
HTML Form URL Encoded (urlencoded-form), 49 bytes
Packets: 418 · Displayed: 2 (0.5%)
Profile: Default
Not Secure
www.demo.amitjakhu.com/login-form/
Login Form
Fill out the form below to login to my super awesome imaginary control panel.
Username
Register
Login

HTTPS


Ethernet: en0
Apply a display filter ... <⌘/>
Expression...
No. Time Source Destination Protocol Length Info
539 12.060370 104.40.147.142 192.168.31.124 TLSv1… 240 Application Data
540 12.060478 192.168.31.124 104.40.147.142 TCP 66 65018 → 443 [ACK] Seq=39 Ack=4925 Win=4090…
541 12.136752 192.168.31.124 8.8.8.8 DNS 75 Standard query 0xea33 A www.twitter.com
542 12.152407 8.8.8.8 192.168.31.124 DNS 121 Standard query response 0xea33 A www.twitt…
543 12.155657 192.168.31.124 63.215.202.158 TCP 54 64018 → 443 [RST, ACK] Seq=1 Ack=1 Win=409…
544 12.155805 192.168.31.124 93.184.220.66 TCP 66 63295 → 443 [FIN, ACK] Seq=1 Ack=1 Win=409…
545 12.156148 192.168.31.124 104.244.42.65 TCP 78 64234 → 443 [SYN] Seq=0 Win=65535 Len=0 MS…
546 12.156288 192.168.31.124 104.244.42.65 TCP 78 64235 → 443 [SYN] Seq=0 Win=65535 Len=0 MS…
547 12.175414 93.184.220.66 192.168.31.124 TCP 60 443 → 63295 [RST] Seq=1 Win=0 Len=0
548 12.185173 104.244.42.65 192.168.31.124 TCP 74 443 → 64235 [SYN, ACK] Seq=0 Ack=1 Win=289…
549 12.185321 192.168.31.124 104.244.42.65 TCP 66 64235 → 443 [ACK] Seq=1 Ack=1 Win=131744 L…
550 12.185816 192.168.31.124 104.244.42.65 TLSv1… 583 Client Hello
551 12.186359 104.244.42.65 192.168.31.124 TCP 74 443 → 64234 [SYN, ACK] Seq=0 Ack=1 Win=289…
552 12.186417 192.168.31.124 104.244.42.65 TCP 66 64234 → 443 [ACK] Seq=1 Ack=1 Win=131744 L…
553 12.186773 192.168.31.124 104.244.42.65 TLSv1… 583 Client Hello
Frame 553: 583 bytes on wire (4664 bits), 583 bytes captured (4664 bits) on interface 0
Ethernet II, Src: Apple_04:bc:28 (c4:2c:03:04:bc:28), Dst: BelkinIn_88:a2:20 (94:10:3e:88:a2:20)
Internet Protocol Version 4, Src: 192.168.31.124, Dst: 104.244.42.65
Transmission Control Protocol, Src Port: 64234, Dst Port: 443, Seq: 1, Ack: 1, Len: 517
Secure Sockets Layer
TLSv1.2 Record Layer: Handshake Protocol: Client Hello
0000 94 10 3e 88 a2 20 c4 2c 03 04 bc 28 08 00 45 00
0010 02 39 00 00 40 00 40 06 00 00 c0 a8 1f 7c 68 f4
0020 2a 41 fa ea 01 bb 7a a6 e1 dc 12 aa 26 0f 80 18
0030 10 15 75 85 00 00 01 01 08 0a 4c 02 66 ed 13 29
0040 c5 8c 16 03 01 02 00 01 00 01 fc 03 03 7b 05 7a
0050 9b 55 b5 f7 f7 75 2c 02 ae 45 86 c1 f4 ea e1 dd
0060 91 59 39 7c 1c 7c ad 8c 19 09 81 08 92 20 b5 8a
0070 6c ee e2 f0 0e aa 16 45 da 21 87 dd a2 62 a9 7e
0080 fe be f6 49 01 13 88 28 a4 79 ed bf b3 49 00 22
0090 ca ca 13 03 13 01 13 02 cc a9 cc a8 c0 2b c0 2f
00a0 c0 2c c0 30 c0 13 c0 14 00 9c 00 9d 00 2f 00 35
00b0 00 0a 01 00 01 91 4a 4a 00 00 ff 01 00 01 00 00
00c0 00 00 14 00 12 00 00 0f 77 77 77 2e 74 77 69 74
Bytes 166-167: Cipher Suite (ssl.handshake.ciphersuite)
Packets: 1943 · Displayed: 1943 (100.0%) · Dropped: 0 (0.0%)
Profile: Default

CLI



tshark - -h

wireshark file extension .pcap

tshark -D

```
iMac27:~ iMac$ tshark -D
1. en0 (Ethernet)
2. fw0 (FireWire)
3. utun0
4. utun1
5. lo0 (Loopback)
6. en1 (Wi-Fi)
7. gif0
8. stf0
9. p2p0
10. EHC250
11. EHC253
12. ciscodump (Cisco remote capture)
13. randpkt (Random packet generator)
14. sshdump (SSH remote capture)
15. udpdump (UDP Listener remote capture)
iMac27:~ iMac$
```

tshark -i en0 **of** tshark -i 1

tshark -i en0 -i any

```
iMac27:~ iMac$ tshark -i en0
Capturing on 'Ethernet'
    1   0.000000 129.143.116.10 → 192.168.31.124 SSL 1514 Continuation Data
    2   0.000121 192.168.31.124 → 129.143.116.10 TCP 54 61743 → 443 [ACK] Seq=1
Ack=1461 Win=8146 Len=0
    3   0.000233 129.143.116.10 → 192.168.31.124 SSL 1514 Continuation Data
    4   0.000344 129.143.116.10 → 192.168.31.124 SSL 1514 Continuation Data
    5   0.000365 192.168.31.124 → 129.143.116.10 TCP 54 61743 → 443 [ACK] Seq=1
Ack=2921 Win=8192 Len=0
    6   0.000466 129.143.116.10 → 192.168.31.124 SSL 1514 Continuation Data
    7   0.000508 192.168.31.124 → 129.143.116.10 TCP 54 61743 → 443 [ACK] Seq=1
Ack=5841 Win=8146 Len=0
    8   0.000646 129.143.116.10 → 192.168.31.124 SSL 1514 Continuation Data
    9   0.000704 192.168.31.124 → 129.143.116.10 TCP 54 61743 → 443 [ACK] Seq=1
Ack=7301 Win=8192 Len=0
   10   0.000716 129.143.116.10 → 192.168.31.124 SSL 1514 Continuation Data
   11   0.000844 129.143.116.10 → 192.168.31.124 SSL 1514 Continuation Data
   12   0.000875 192.168.31.124 → 129.143.116.10 TCP 54 61743 → 443 [ACK] Seq=1
Ack=10221 Win=8146 Len=0
   13   0.040324 129.143.116.10 → 192.168.31.124 SSL 1514 Continuation Data
   14   0.040468 192.168.31.124 → 129.143.116.10 TCP 54 61743 → 443 [ACK] Seq=1
Ack=11681 Win=8192 Len=0
   15   0.045303 129.143.116.10 → 192.168.31.124 SSL 1514 Continuation Data
   16   0.045482 129.143.116.10 → 192.168.31.124 SSL 1514 Continuation Data
   17   0.045541 192.168.31.124 → 129.143.116.10 TCP 54 61743 → 443 [ACK] Seq=1
Ack=14601 Win=8146 Len=0
   18   0.045603 129.143.116.10 → 192.168.31.124 SSL 1514 Continuation Data
   19   0.045657 192.168.31.124 → 129.143.116.10 TCP 54 61743 → 443 [ACK] Seq=1
Ack=16061 Win=8192 Len=0
   20   0.045715 129.143.116.10 → 192.168.31.124 SSL 1514 [TCP Previous segment
not captured] , Continuation Data
   21   0.045753 192.168.31.124 → 129.143.116.10 TCP 66 [TCP Dup ACK 19#1] 61743
 → 443 [ACK] Seq=1 Ack=16061 Win=8192 Len=0 SLE=17521 SRE=18981
   22   0.045889 129.143.116.10 → 192.168.31.124 SSL 1514 Continuation Data
   23   0.045923 192.168.31.124 → 129.143.116.10 TCP 66 [TCP Dup ACK 19#2] 61743
 → 443 [ACK] Seq=1 Ack=16061 Win=8192 Len=0 SLE=17521 SRE=20441
   24   0.081958 129.143.116.10 → 192.168.31.124 SSL 1514 Continuation Data
   25   0.082045 192.168.31.124 → 129.143.116.10 TCP 66 [TCP Dup ACK 19#3] 61743
 → 443 [ACK] Seq=1 Ack=16061 Win=8192 Len=0 SLE=17521 SRE=21901
   26   0.095008 129.143.116.10 → 192.168.31.124 TCP 1514 443 → 61743 [ACK] Seq=
21901 Ack=1 Win=22 Len=1460
```

tshark *luistert enkel naar interface 1*

tshark -i en0 -w /home/iMac27/testCap.pcap

beter

tshark -i en0 -w /tmp/testCap.pcap

```
[iMac27:~ iMac$ tshark -i en0  -w /home/iMac27/testCap.pcap
Capturing on 'Ethernet'
tshark: The file to which the capture would be saved ("/home/iMac27/tes
") could not be opened: No such file or directory.

[iMac27:~ iMac$ tshark -i en0  -w /tmp/testCap.pcap
Capturing on 'Ethernet'
8248 ^C
iMac27:~ iMac$
```

iMac — g?|vx{1??/?ฑlo?}?T? — -bash — 80×41

```
[iMac27:~ iMac$ cat /tmp/testCap.pcap
?M<+?????????=Intel(R) Core(TM) i3 CPU         550  @ 3.20GHz (with SSE4.2)-Mac O
S X 10.13.6, build 17G65 (Darwin 17.7.0).Dumpcap (Wireshark) 2.6.3 (v2.6.3-0-ga6
2e6c27)?en0
                -Mac OS X 10.13.6, build 17G65 (Darwin 17.7.0)\?Ywqq?,?(?>??Ec??
@5E??Re??|??Q??r쌋?{??
?bL9l?*=??H???_y        ?^??<?;?6??&???????9B!9i?dYwcBB?>??  ?,?E4@@??|?Re??r쌋Q?
??l[
L9l??bd?Yw?S?qq?,?(?>??Ec??@5E??Re??|??Q?r쌋?{
?bL9l?*=??H???  B?V$Xj?3s?????V??T???6?O?r`?dYw?S?BB?>??  ?,?E4@@??|?Re??r쌋Q?C??l
[
L9l??bd?Yw?\?qq?,?(?>??Ec??@5E??Re??|??Q?Cr쌋?{??
?bL9l?*=??H???m?]?LA?_??D?FjE%?xg??.C??_???dYw-]?BB?>??  ?,?E4@@??|?Re??r쌋Q?r??l
[
L9l??bd?YwZ??qq?,?(?>??Ec??@5E??Re??|??Q?rr쌋?{
?bL9l?*=??H???1&?=g?VFQ???k?R????
                                Ud?????&ej???dYw???BB?>??  ?,?E4@@??|?Re??r쌋QC?
?l[
L9l??bd?Yw???qq?,?(?>??Ec??@5E??Re??|??QCr쌋?{?
?bL9l?*=??H??ᴿ??)??0???"?ъ‖c*?i6D??Ω?>?dYw???BB?>??  ?,?E4@@??|?Re??r쌋Q?È?l[
L9l??bd?YwM?qq?,?(?>??Ec??@5E??Re??|??Q??r쌋?{`v
?bL9l?*=??H????PU~S}h2q???-?k;?*~d?'7???}???9?dYw??BB?>??  ?,?E4@@??|?Re??r쌋Q???
?l[
L9l??bd?Yw!?qq?,?(?>??Ec??@5E??Re??|??Q??r쌋?{?W
I?|??K????????e?Oi?dYw??BB?>??  ?,?E4@@??|?Re??r쌋Q?.??l[
L9l??d?Yw??????,?(?>??E???@5EG?Re??|??Q?.r쌋?{8
?L9l??=??H???c?)?"s???#?鐮>?k?;(}<R????)??z??./@'?P?m??"?(ћr?^?i???R??h?yp??  G?x
????HE???&??#bp?#????p?n?=Y?6>BF??+?dYwZ??BB?>??  ?,?E4@@??|?Re??r쌋QO9?l[
L9l??d8Yw????>??  ?,?E@@??|?Re??r쌋QO9m.
?]?n&???(?????9???¦?<?  ?
?h??~?=?n??>?P?D???SH???aa#?DJa?Nl????+ŽJiW???Q???N??????9?5??&??Y?ъ??A?u1
??j???ǯ??-8\Yw?
                ?<<??????#$*#$*?=?????F\dYw=<?BB?,?(?>??E4??@5EO9Re??|??QOy?V?{??
?b
  L9l?d|Ywa??\\?,?(?>??EN??@5D??Re??|??QOy?V?{?
?bCL9l?=??H????M(N?P?Y(??HwKL???
??F???$?????þ2?@?v??C3%m5SN1?
???Yθ?+A?f???a?A6?^??l?Ɪ.t???M?0(?US??Ro?9y?r??{?S?1??h
8:?.vN                                                          ?{
```

testCap.pcap

Apply a display filter ... <⌘/> Expression...

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	185.82.210.189	192.168.31.124	TLSv1…	113	Application Data
2	0.000094	192.168.31.124	185.82.210.189	TCP	66	59577 → 443 [ACK] Seq=1 Ack=48 Win=4094 Le…
3	0.014471	185.82.210.189	192.168.31.124	TLSv1…	113	Application Data
4	0.014583	192.168.31.124	185.82.210.189	TCP	66	59577 → 443 [ACK] Seq=1 Ack=95 Win=4094 Le…
5	0.016845	185.82.210.189	192.168.31.124	TLSv1…	113	Application Data
6	0.016936	192.168.31.124	185.82.210.189	TCP	66	59577 → 443 [ACK] Seq=1 Ack=142 Win=4094 L…
7	0.030293	185.82.210.189	192.168.31.124	TLSv1…	113	Application Data
8	0.030396	192.168.31.124	185.82.210.189	TCP	66	59577 → 443 [ACK] Seq=1 Ack=189 Win=4094 L…
9	0.037518	185.82.210.189	192.168.31.124	TLSv1…	113	Application Data
10	0.037619	192.168.31.124	185.82.210.189	TCP	66	59577 → 443 [ACK] Seq=1 Ack=236 Win=4094 L…
11	0.051272	185.82.210.189	192.168.31.124	TLSv1…	113	Application Data
12	0.051368	192.168.31.124	185.82.210.189	TCP	66	59577 → 443 [ACK] Seq=1 Ack=283 Win=4094 L…
13	0.054300	185.82.210.189	192.168.31.124	TLSv1…	113	Application Data
14	0.054401	192.168.31.124	185.82.210.189	TCP	66	59577 → 443 [ACK] Seq=1 Ack=330 Win=4094 L…
15	0.057843	185.82.210.189	192.168.31.124	TLSv1…	205	Application Data

```
▶ Frame 1: 113 bytes on wire (904 bits), 113 bytes captured (904 bits) on interface 0
▶ Ethernet II, Src: BelkinIn_88:a2:20 (94:10:3e:88:a2:20), Dst: Apple_04:bc:28 (c4:2c:03:04:bc:28)
▶ Internet Protocol Version 4, Src: 185.82.210.189, Dst: 192.168.31.124
▶ Transmission Control Protocol, Src Port: 443, Dst Port: 59577, Seq: 1, Ack: 1, Len: 47
▶ Secure Sockets Layer
```

```
0000  c4 2c 03 04 bc 28 94 10  3e 88 a2 20 08 00 45 00   ·,···(·· >·· ··E·
0010  00 63 93 b6 40 00 35 06  45 aa b9 52 d2 bd c0 a8   ·c··@·5· E··R····
0020  1f 7c 01 bb e8 b9 1f 51  cf e5 72 ec 8d 83 80 18   ·|·····Q ··r·····
0030  01 7b 85 a4 00 00 01 01  08 0a c7 62 1b fa 4c 39   ·{····· ···b··L9
0040  6c 8d 17 03 03 00 2a 19  3d af 81 48 e2 c3 c7 5f   l·····*· =··H···_
0050  79 1b 50 09 fd 5e 7f d7  fa 10 3c 89 3b a5 36 17   y·P··^·· ··<·;·6·
0060  d9 f5 26 b6 a8 f9 85 fa  b3 08 ca e4 39 42 21 39   ··&····· ····9B!9
0070  69                                                i
```

testCap.pcap Packets: 8248 · Displayed: 8248 (100.0%) Profile: Default

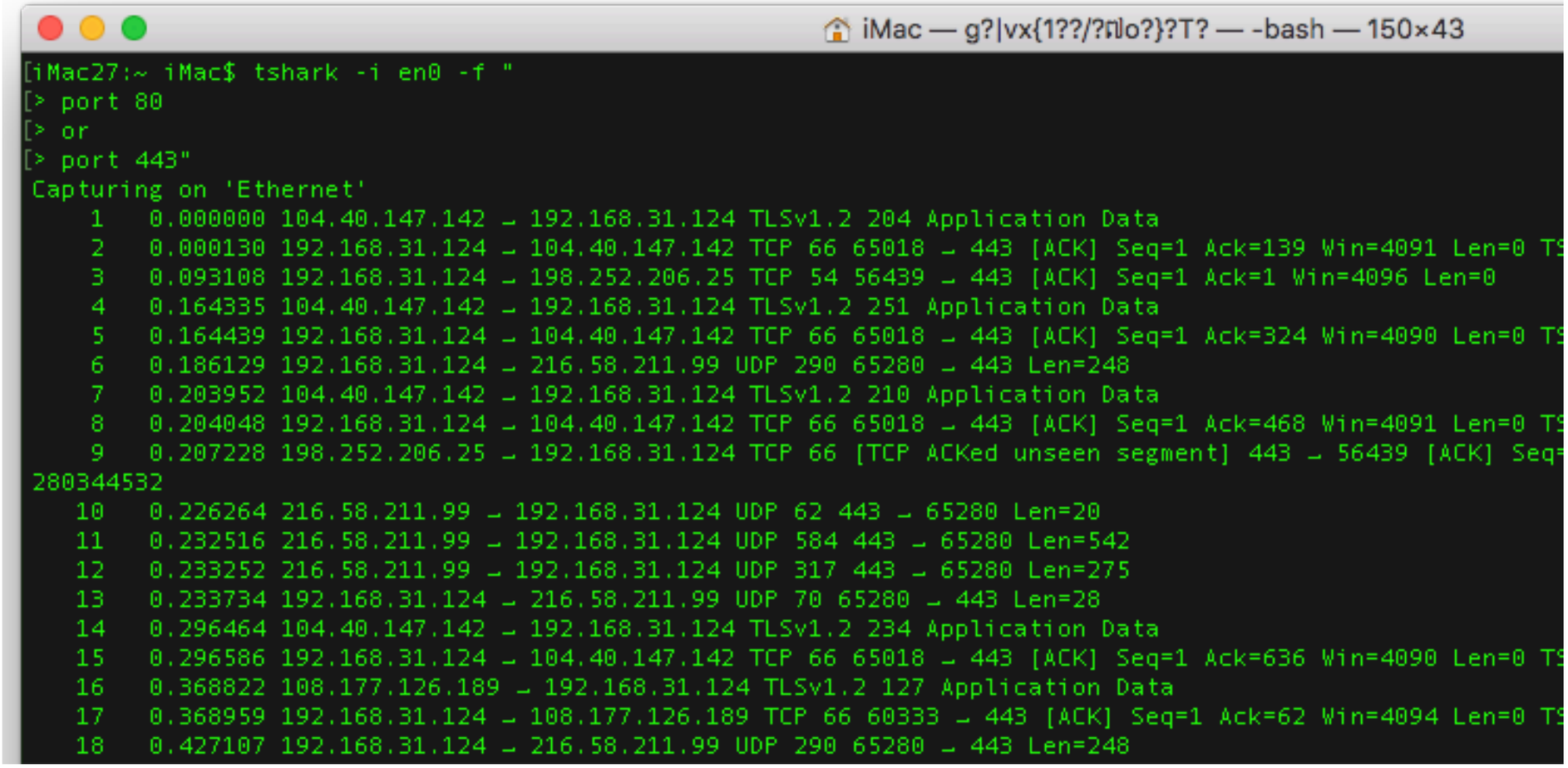
tshark i en0 -a duration: 250 -w /tmp/file.pcap *stopt na 250 seconden*

tshark i en0 -a filesize: 250 -w /tmp/file.pcap *stopt als de file 250KB groot is*

tshark i en0 f “port 80 or port 443 or port 53”-b filesize: 5 -a files: 10 -w /tmp/file.pcap *luistert naar en0 poorten 80,443,53 en stopt als het 10 files van 5KB aangemaakt heeft*

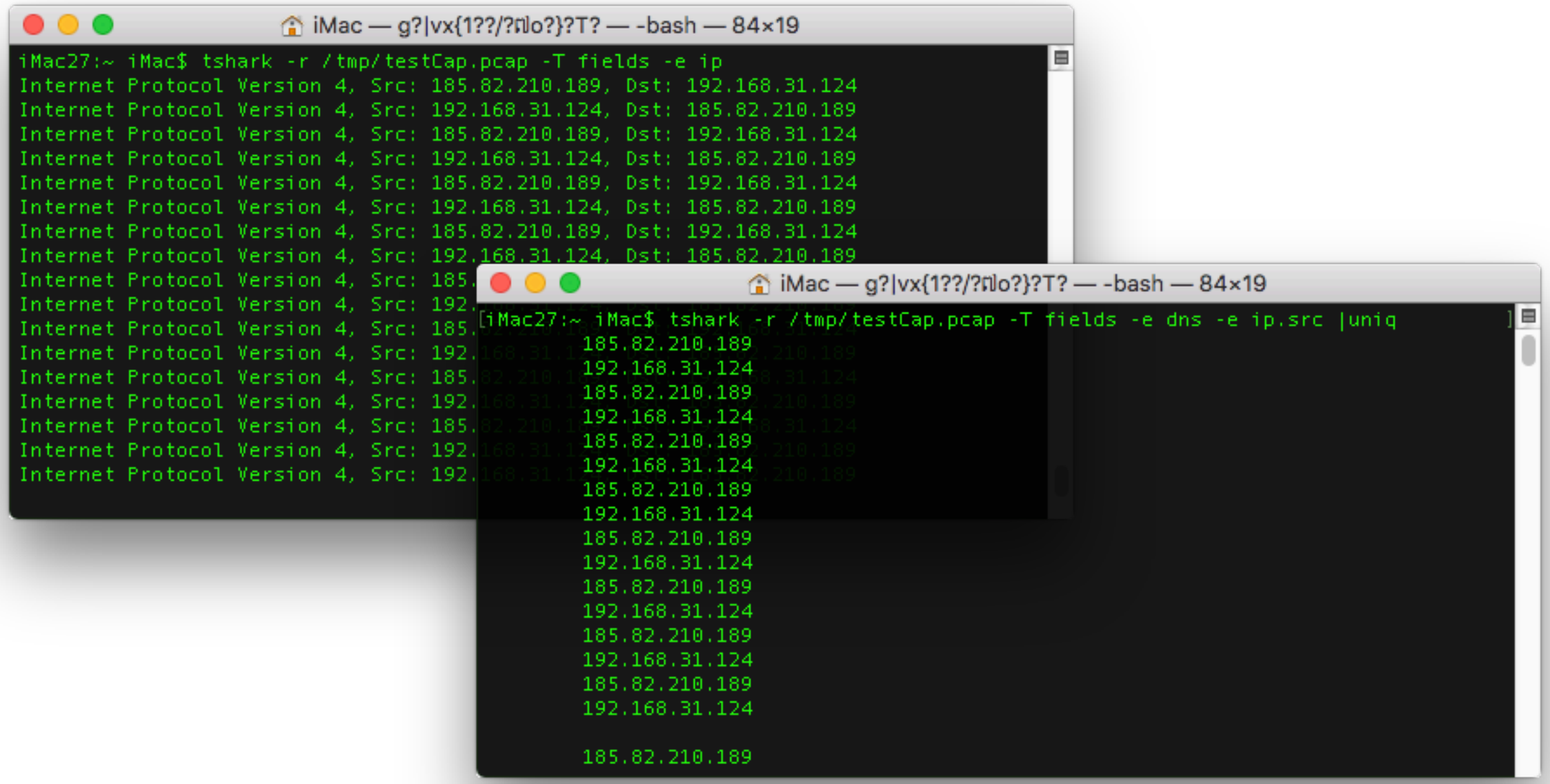
Capture filters =/= Display filters

tshark i en0 -f = filter



Filter file

tshark -r /tmp/testCap.pcap -T fields -e ip




```
tshark -r /tmp/testCap.pcap -T fields -e dns -e ip.src |uniq
```

Master Mode : alle traffic passeert via jou (bv router)
Ad-hoc Mode: p2p
Mash: via andere computers
Repeater Mode: alles wat binnenkomt wordt gerepeat
Monitor Mode: alle traffic wordt onderschept

check compatibiliteit netwerkkaart:
https://www.aircrack-ng.org/doku.php?id=compatibility_drivers
indien nodig usb netwerkkaart

```
RX errors 0  dropped 0  overruns 0  frame 0
TX packets 265617  bytes 28110199 (26.8 MiB)
TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[root@localhost ~]# iwconfig wlp2s0 | grep -i mode
          Mode:Managed  Frequency:2.437 GHz  Access Point: 90:F6:52:C1:BB:18
[root@localhost ~]#
```

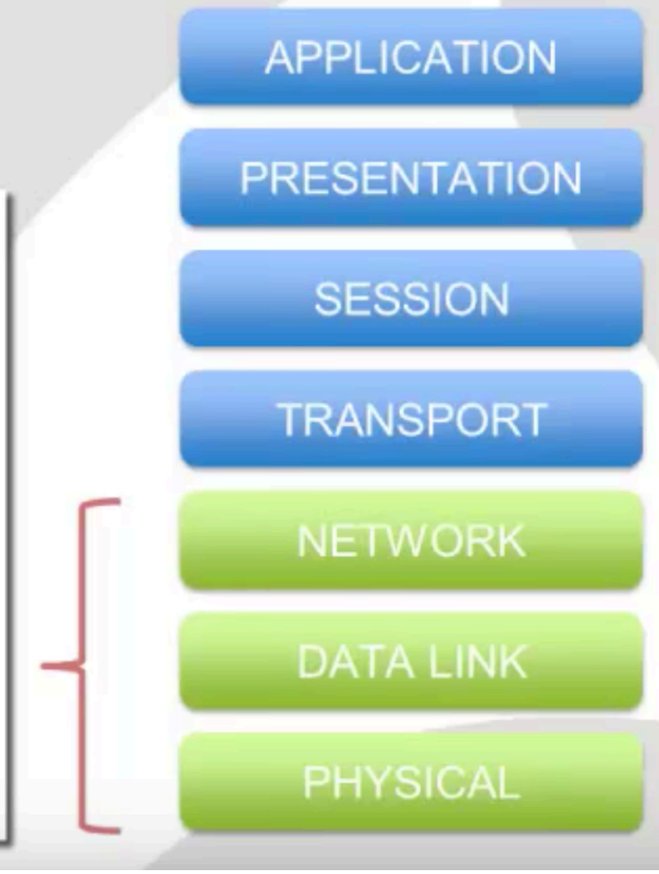
```
[root@localhost ~]# iwconfig wlp2s0 mode monitor
Error for wireless request "Set Mode" (8B06) :
    SET failed on device wlp2s0 ; Device or resource busy.
[root@localhost ~]# ifconfig wlp2s0 down
```

```
[root@localhost ~]# iwconfig wlp2s0 mode monitor
[root@localhost ~]# ifconfig wlp2s0 up
[root@localhost ~]#
```

```
[root@localhost ~]# iwconfig wlp2s0
wlp2s0    IEEE 802.11bgn  Mode:Monitor
```

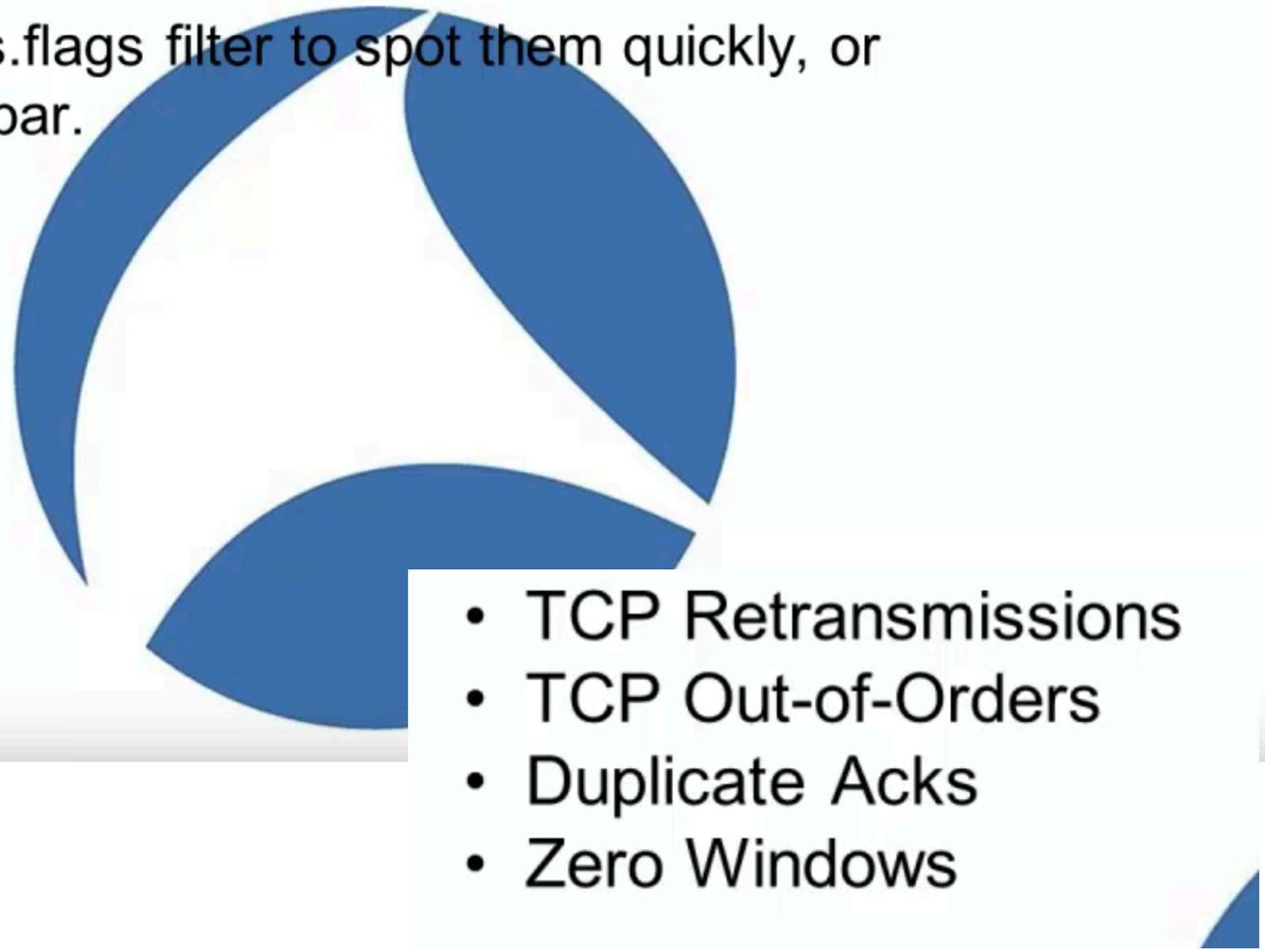
When a problem strikes

- Network engineers check network interfaces, utilization levels, link errors and the wireless environment
- They want to prove it's not the network.



What should I look for in traces?

- Wireshark has some great error events to flag TCP problems.
- Use the tcp.analysis.flags filter to spot them quickly, or the intelligent slide bar.



- TCP Retransmissions
- TCP Out-of-Orders
- Duplicate Acks
- Zero Windows

Wireshark

Apply a display filter ... <⌘/>

Expression...

No.	Time	DTime	Source	Destination	Protocol	Length	Info
1	0.000000	0.000000	192.168.31.121	224.0.0.2	IGMPv2	60	Membership Report group 224.0.0.2
2	0.036044	0.036044	G-ProCom_2a:d0:3d	Broadcast	ARP	60	Who has 192.168.31.64? Tell 192.168.31
3	0.051470	0.015426	G-ProCom_2a:d0:3d	Broadcast	ARP	60	Who has 192.168.31.70? Tell 192.168.31
4	0.128241	0.076771	192.168.31.122	224.0.0.22	IGMPv2	60	Membership Report group 224.0.0.22
5	0.128549	0.000308	192.168.31.121	224.0.0.22	IGMPv2	60	Membership Query, specific for group 2
6	0.128552	0.000003	192.168.31.121	224.0.1.60	IGMPv2	60	Membership Query, specific for group 2
7	0.129208	0.000656	192.168.31.1	224.0.0.22	IGMPv2	60	Membership Report group 224.0.0.22
8	0.180513	0.051305	ZyxelCom_5e:45:09	Broadcast	0x8899	60	Realtek Layer 2 Protocols
9	0.304923	0.124410	192.168.31.124	89.207.16.157	TCP	54	50904 → 443 [RST, ACK] Seq=1 Ack=1 Win
10	0.305951	0.001028	192.168.31.124	8.8.8.8	DNS	83	Standard query 0xa8b2 A col.eum-appdyn
11	0.332208	0.0262					
12	0.336292	0.0040					l.e
13	0.336910	0.0006					en=
14	0.369040	0.0321					Win
15	0.369191	0.0001					317

▶ Frame 1: 60 bytes on wi
▶ Ethernet II, Src: Shenz
▶ Internet Protocol Versi
▶ Internet Group Manageme

```
0000  01 00 5e 00 00 02 40
0010  00 20 2f 21 00 00 01
0020  00 02 94 04 00 00 16
0030  6f 00 80 80 84 00 80
```

Ready to load or capture

file: Default

Wireshark · Preferences

- ▼ Appearance
  - Columns
  - Font and Colors
  - Layout
- Capture
- Expert
- Filter Buttons
- Name Resolution
- ▶ Protocols
- ▶ Statistics
- Advanced

Displayed	Title	Type	Fields	Field
☑	No.	Number		
☑	Time	Time (format as specified)		
☑	DTime	Delta time displayed		
☑	Source	Source address		
☑	Destination	Destination address		
☑	Protocol	Protocol		
☑	Length	Packet length (bytes)		
☑	Info	Information		

+ −

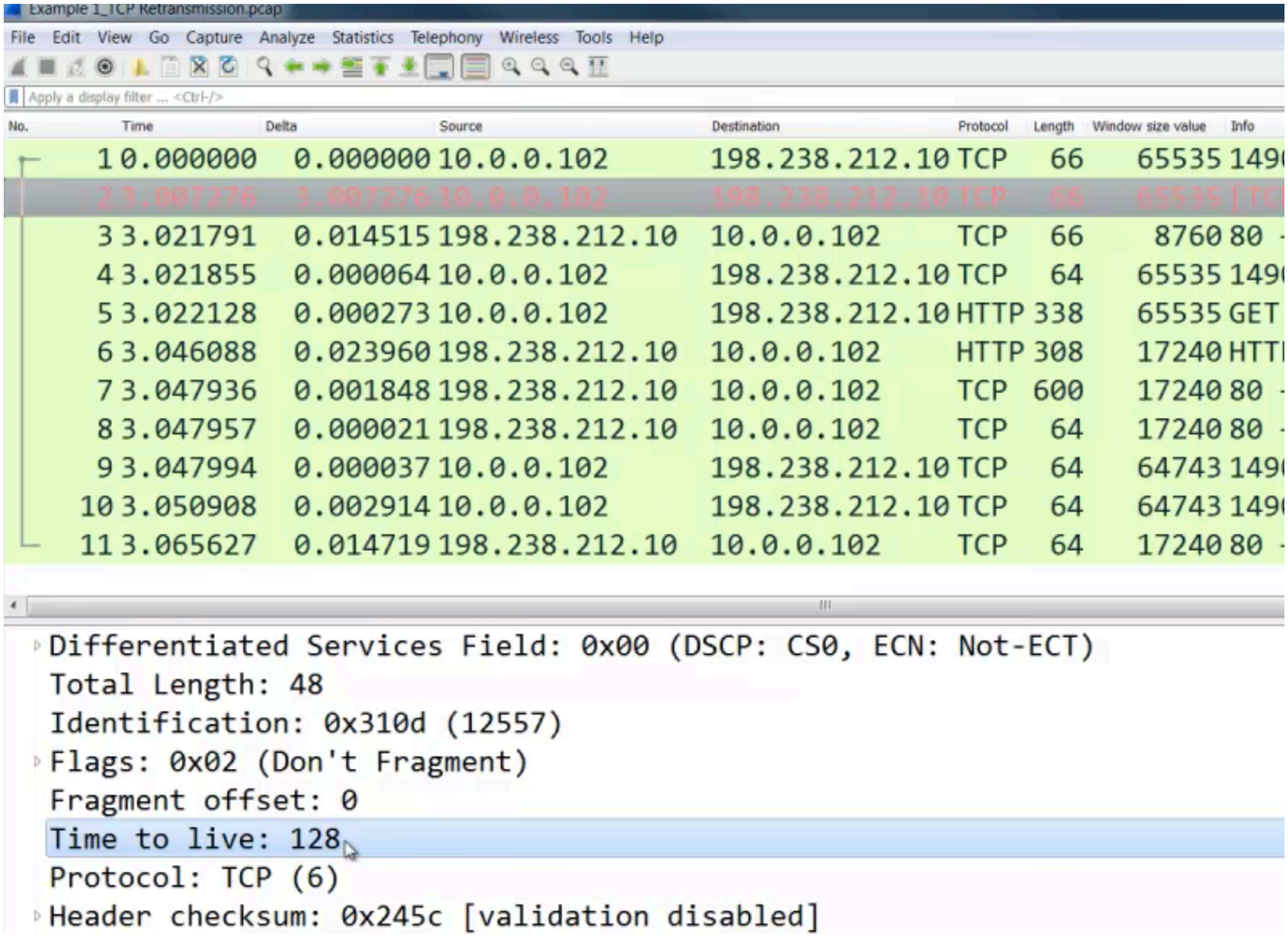
Help

Cancel OK

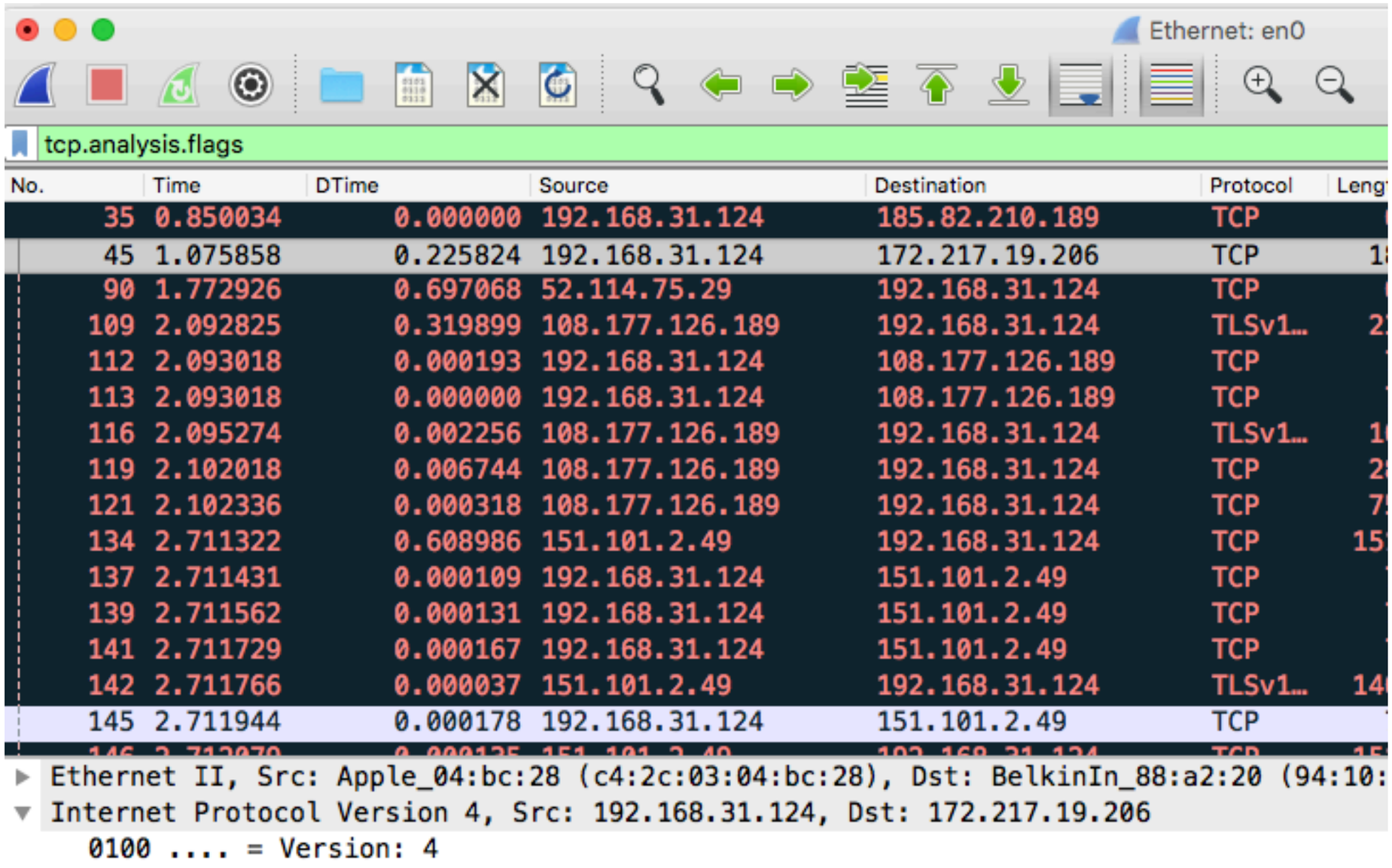
vives

VIVES HBO V

Packet not routed - elke hop TTL -1



Filteren op bad TCP



Retransmission: opnieuw zenden van packets
Out Of order: volgorde packets reversed of packet not captured Dup Ack = Hey server, I'm missing a packet!
KeepAlive = Hey, leef je nog?

Meer informatie:

https://www.youtube.com/watch?v=15wDU3Wx1h0

Wireshark oefeningen:

Les 3 Wireshark oefeningen.pdf