

# Mastering Filtering



**Sake Blok**  
*Relational therapist for computer systems*  
sake.blok@SYN-bit.nl



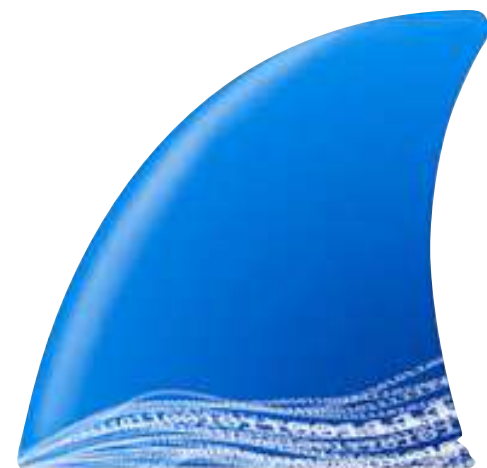


# Agenda

- Using the filter bar
- Right-click / Drag & drop filtering
- Filtering from dialogs
- Display filter buttons
- Multiple occurrences of fields
- Multiple protocol layers
- Further reading / viewing
- Q&A



# About me...







**SYN-bit**  
deep traffic analysis

**Application and network troubleshooting**

---

**Protocol and packet analysis**

---

**Training (Wireshark, TCP, SSL)**

**[www.SYN-bit.nl](http://www.SYN-bit.nl)**



# Using the filter bar

- Filter on...
  - protocol presence
  - field presence
- Using basic operators
  - == != < > <= >=
- Set filters
  - tcp.port in {80, 8000..8099}
- String filter
  - http.cookie contains "PHPSESSION"
- Combine with and, or, not, (...)





# DEMO 1



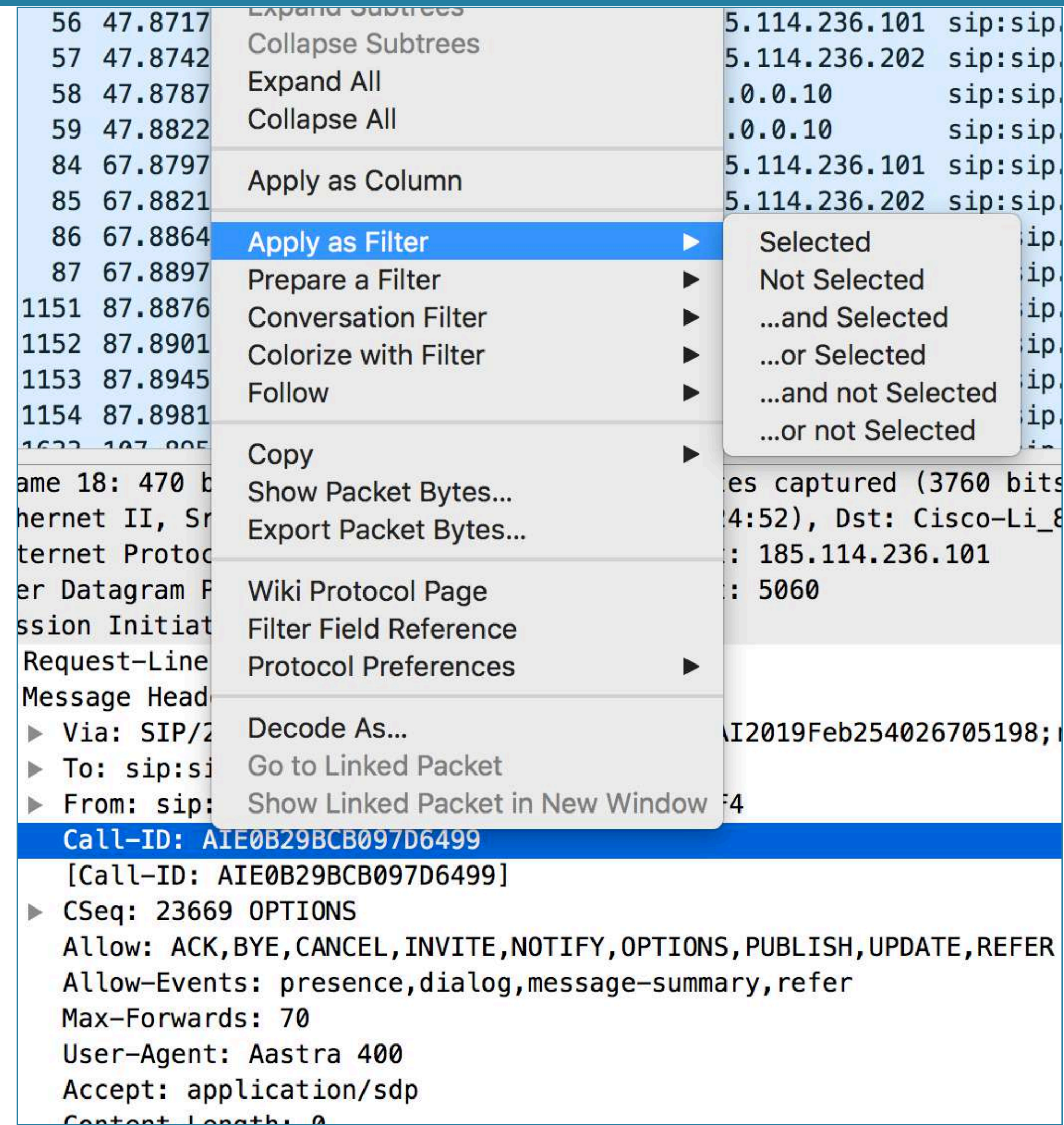
# Filter operators

C-like	English	Description	Example
<code>==</code>	<code>eq</code> , <i><b>any_eq</b></i>	Equals	<code>ip.addr == 192.168.1.1</code>
<code>!=</code>	<code>ne</code> , <i><b>all_ne</b></i>	(All) Not equals	<code>ip.addr != 192.168.1.1</code>
<code>===</code>	<i><b>all_eq</b></i>	<i><b>All equals</b></i>	<i><b>ip.addr === 192.168.1.1</b></i>
<code>!==</code>	<i><b>any_ne</b></i>	<i><b>Any not equals</b></i>	<i><b>ip.addr !== 192.168.1.0/24</b></i>
<code>&gt;</code>	<code>gt</code>	Greater than	<code>dns.time &gt; 0.1</code>
<code>&lt;</code>	<code>lt</code>	Less than	<code>frame.len &lt; 60</code>
<code>&gt;=</code>	<code>ge</code>	Greater than or equal to	<code>frame.number &gt;= 1000</code>
<code>&lt;=</code>	<code>le</code>	Less than or equal to	<code>frame.number &lt;= 2000</code>
	<code>contains</code>	Protocol, field or slice contains	<code>http.host contains "amazon"</code>
<code>~</code>	<code>matches</code>	Matches regular expression	<code>http.host matches "^www\\.\\.\\.\\.\\.nl\$"</code>
<code>&amp;</code>	<code>bitwise_and</code>	Bitwise and on the field	<code>tcp.flags&amp;7, <b>tcp.flags&amp;18==2</b></code>
	<code>in</code>	Set membership	<code>tcp.port in {443 80..89} <b>tcp.port in {443, 80..89}</b></code>
<code>!</code>	<code>not</code>	Logical NOT	<code>not ip.addr == 192.168.1.1</code>
<code>  </code>	<code>or</code>	Logical OR	<code>udp.port==53 or tcp.port==53</code>
<code>&amp;&amp;</code>	<code>and</code>	Logical AND	<code>ip.addr == 10.0.0.1 and udp.port==5060</code>
<code>( )</code>		Group expressions together	<code>ip.addr == 10.0.0.1 and (icmp or dns)</code>



# Right-click / Drag & drop filtering

- Right-click on any item in...
  - Packet list
  - Packet details
- Or drag & drop any item to the filter bar
- Replace filter
  - ...or extend it
- Apply filter (default)
  - ... or prepare (hold shift before drop)



The screenshot shows a network analysis tool interface. On the left, a packet list is visible with columns for packet number, time, and IP addresses. A context menu is open over the list, with 'Apply as Filter' selected. A sub-menu is also open, showing various filter options. Below the packet list, a packet detail pane is visible, showing the contents of a selected packet, including headers and body text.

Packet No.	Time	IP	Protocol
56	47.8717	5.114.236.101	sip:sip
57	47.8742	5.114.236.202	sip:sip
58	47.8787	.0.0.10	sip:sip
59	47.8822	.0.0.10	sip:sip
84	67.8797	5.114.236.101	sip:sip
85	67.8821	5.114.236.202	sip:sip
86	67.8864		ip
87	67.8897		ip
1151	87.8876		ip
1152	87.8901		ip
1153	87.8945		ip
1154	87.8981		ip

Context Menu Options:

- Expand Subtrees
- Collapse Subtrees
- Expand All
- Collapse All
- Apply as Column
- Apply as Filter** (selected)
  - Selected
  - Not Selected
  - ...and Selected
  - ...or Selected
  - ...and not Selected
  - ...or not Selected
- Prepare a Filter
- Conversation Filter
- Colorize with Filter
- Follow
- Copy
- Show Packet Bytes...
- Export Packet Bytes...
- Wiki Protocol Page
- Filter Field Reference
- Protocol Preferences
- Decode As...
- Go to Linked Packet
- Show Linked Packet in New Window



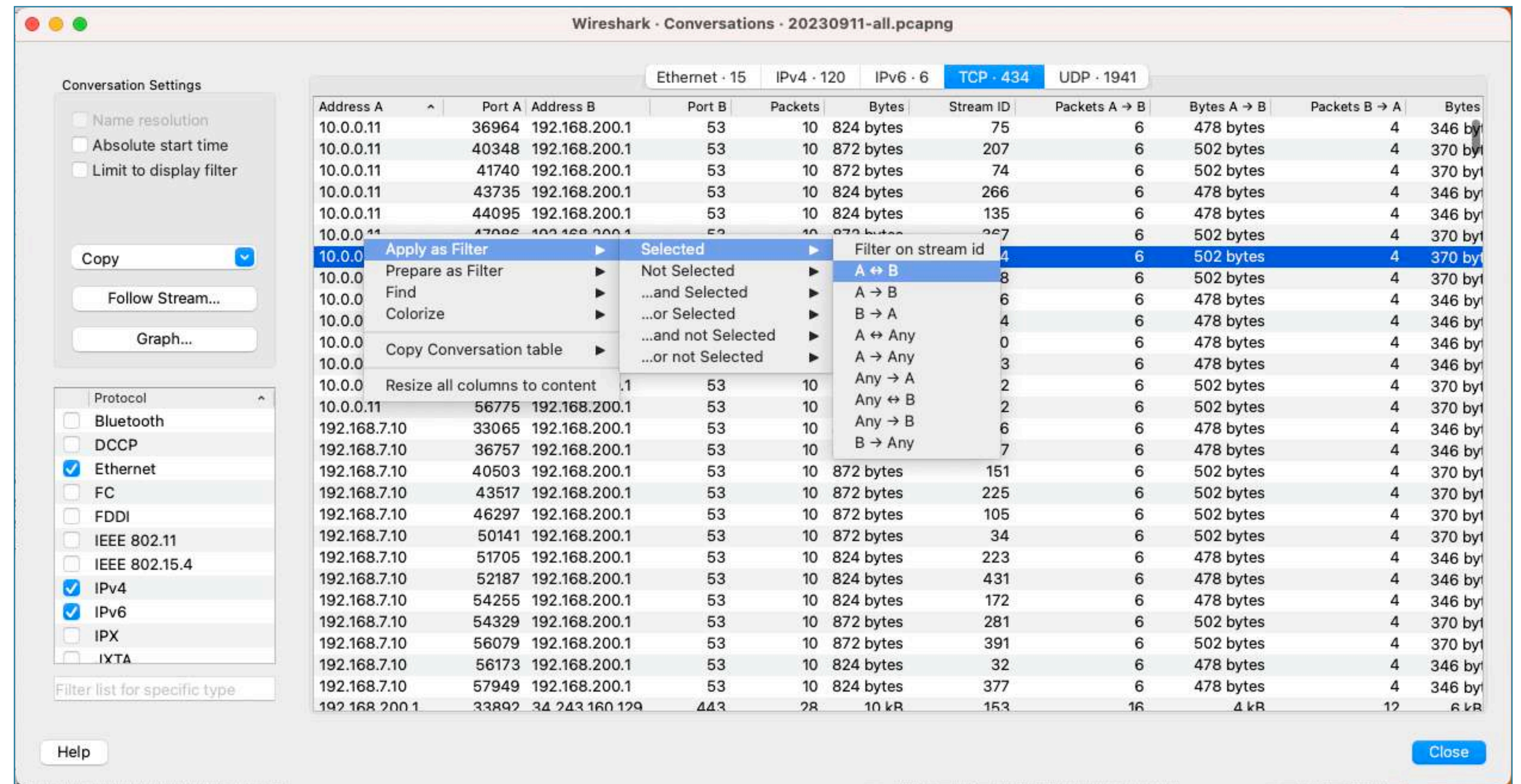
# DEMO 2





# Filtering from dialogs

- Rightclick on items in...
  - Endpoints
  - Conversations
  - Protocol hierarchy
  - etc.





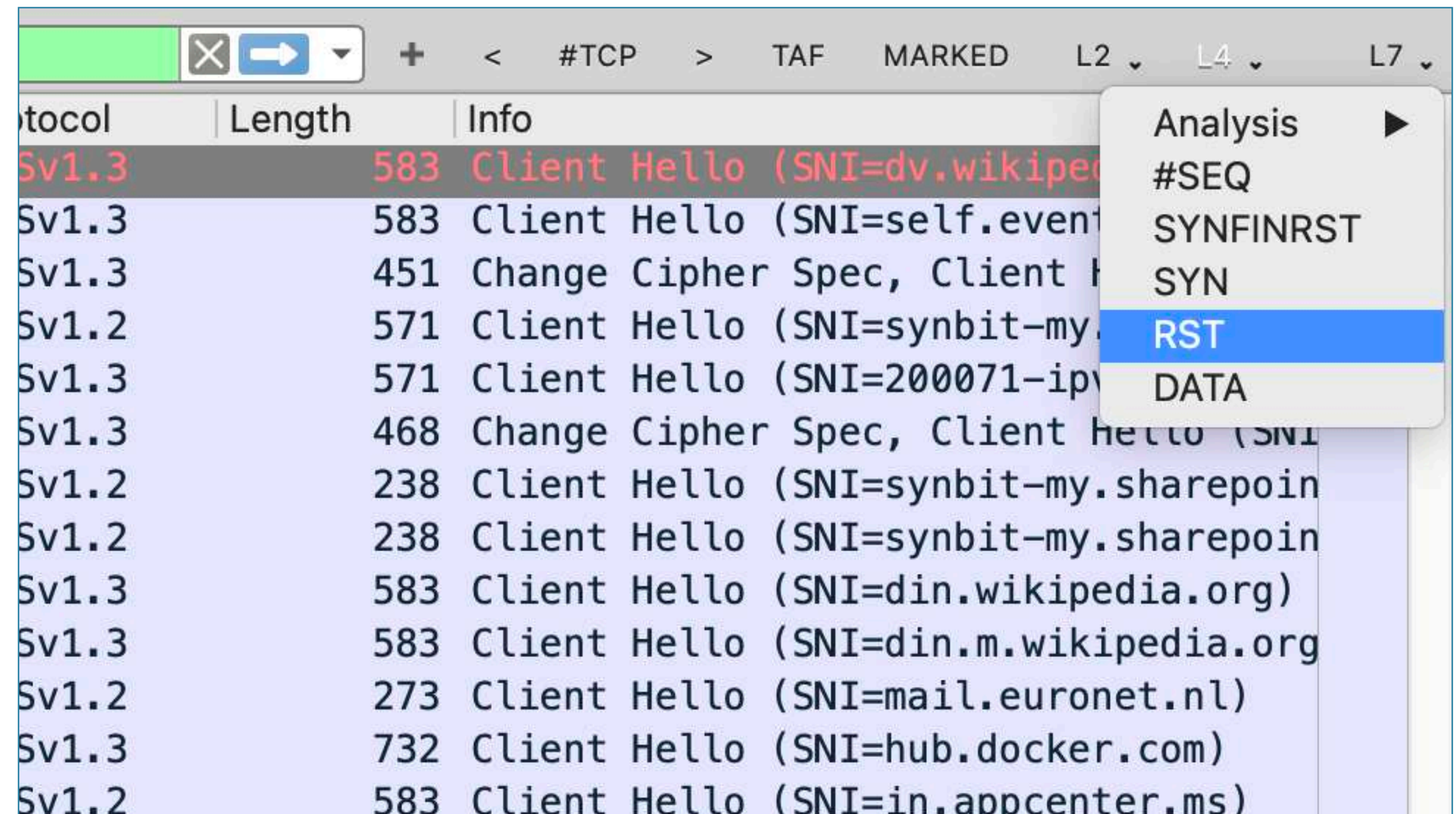
# DEMO 3





# Display filter buttons

- Handy 1 click filter actions
  - use + next to filter bar to create
- ...or more with dropdowns
  - use // in the filter button name
- Field references are powerful
  - `tcp.stream == ${tcp.stream}`
  - `ip.addr == ${dns.a}`



# DEMO 4





# Multiple occurrences of fields

- Implied **any** on **\*all\*** operators
  - ip.addr == 192.168.1.1
- ...except for implied **all** for **!=**
  - ip.addr != 192.168.1.1
- Implied **any** on **!==**
  - ip.dsfield.dscp != 0
- Overrule with explicit **all** or **any**
  - all tcp.port >= 1024
  - all udp.port == 53



# DEMO 5





# Multiple protocol layers

- Specify individual protocol layers with #n
  - ip.addr#2 == 192.168.1.1
  - vlan.id#2 == 10
  - vlan.id#-1 == 10





# Filtering on specific protocol layers

```

▶ Frame 2416: 94 bytes on wire (752 bits), 94 bytes captured (752 bits) on interface en0, id 0
▶ Ethernet II, Src: Ubiquiti 9b:fe:a6 (24:5a:4c:9b:fe:a6), Dst: Apple_5f:32:bb (08:f8:bc:5f:32:bb)
▼ Internet Protocol Version 4, Src: 192.168.3.1, Dst: 192.168.3.163
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0xc0 (DSCP: CS6, ECN: Not-ECT)
  Total Length: 80
  Identification: 0x0f2e (40766)
  ▶ 000. .... =
  ...0 0000 0
  Time to Live: 64
  Protocol: ICMP (1)
  Header Checksum: 0x52ba [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 192.168.3.1
  Destination Address: 192.168.3.163
▼ Internet Control Message Protocol
  Type: 11 (Time-to-live exceeded)
  Code: 0 (Time to live exceeded in transit)
  Checksum: 0xba7d [correct]
  [Checksum Status: Good]
  Unused: 00000000
▼ Internet Protocol Version 4, Src: 192.168.3.163, Dst: 1.0.0.1
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 52
  Identification: 0xd38c (54156)
  ▶ 000. .... =
  ...0 0000 0
  Time to Live:
  Protocol: UDP (17)
  Header Checksum: 0x20e1 [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 192.168.3.163
  Destination Address: 1.0.0.1
  ▶ [Destination GeoIP: Melbourne, AU, ASN 13335, CLOUDFLARENET]
▶ User Datagram Protocol, Src Port: 54155, Dst Port: 33435
▶ Data (24 bytes)
  
```

First IP layer (ip#1)

Second IP layer (ip#2)

Field	192.168.3.1	192.168.3.163	1.0.0.1
ip.addr	✓	✓	✓
ip.addr#1	✓	✓	
ip.addr#2		✓	✓
ip.src	✓	✓	
ip.src#1	✓		
ip.src#2		✓	
ip.dst		✓	✓
ip.dst#1		✓	
ip.dst#2			✓



# DEMO 6



# Further reading / viewing

- **Wireshark Users Guide**

- [https://www.wireshark.org/docs/wsug\\_html\\_chunked/ChWorkDisplayFilterSection.html](https://www.wireshark.org/docs/wsug_html_chunked/ChWorkDisplayFilterSection.html)

- **Wireshark Wiki**

- <https://wiki.wireshark.org/DisplayFilters>

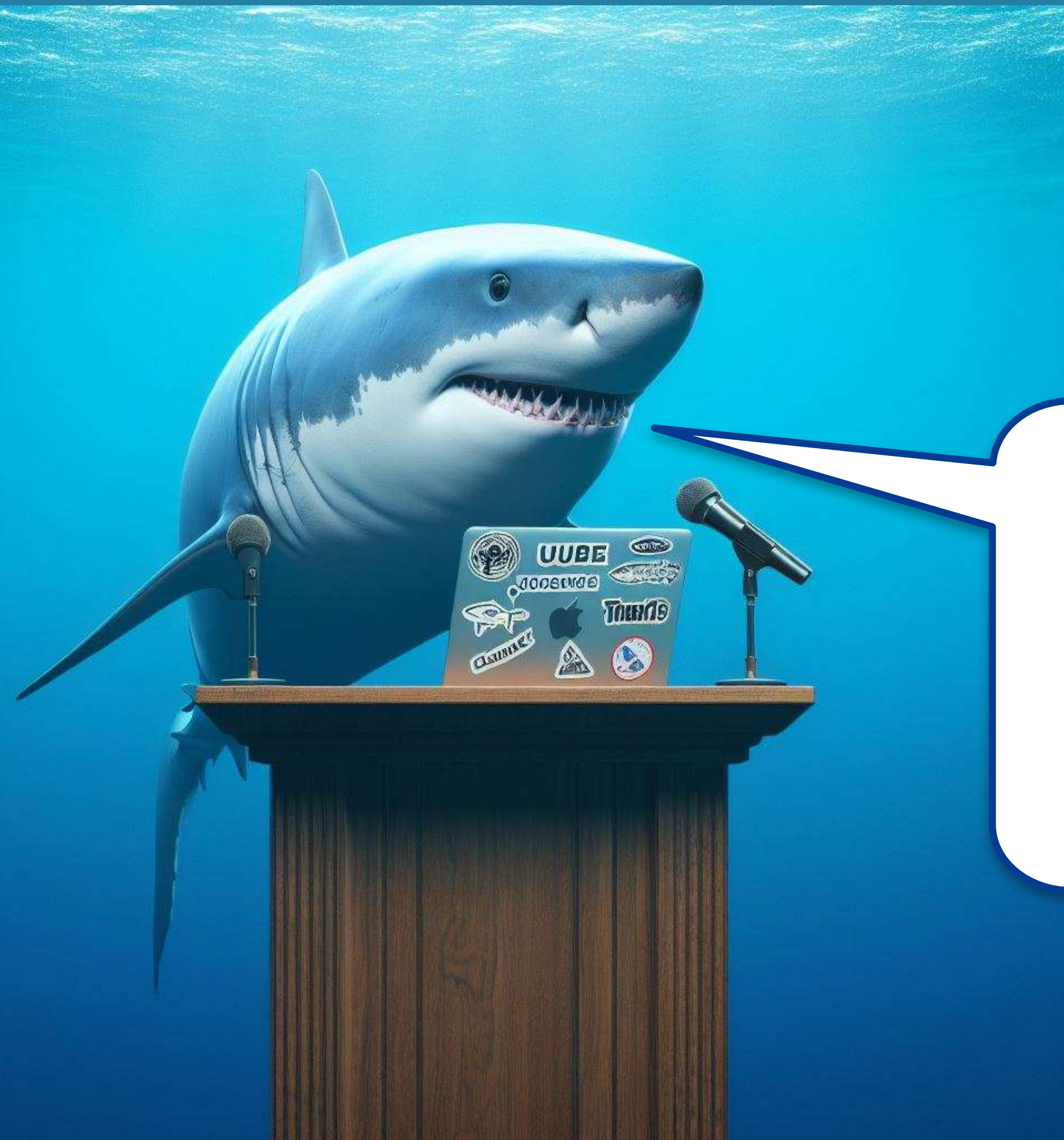
- **Wireshark display filter reference**

- <https://www.wireshark.org/docs/dfref/>

- **Sharkfest retrospective pages**

- <https://sharkfest.wireshark.org/retrospective/>





**Time for Q & A**

# FIN/ACK/FIN/ACK

*Still questions?*  
*sake.blok@SYN-bit.nl*



**SYN-bit**  
deep traffic analysis