IPv6/6LoWPAN with Wireshark

March 2016 – ICTP

Alvaro Vives (alvaro.vives@nodo6.com)
NODO6 (www.nodo6.com)

Content

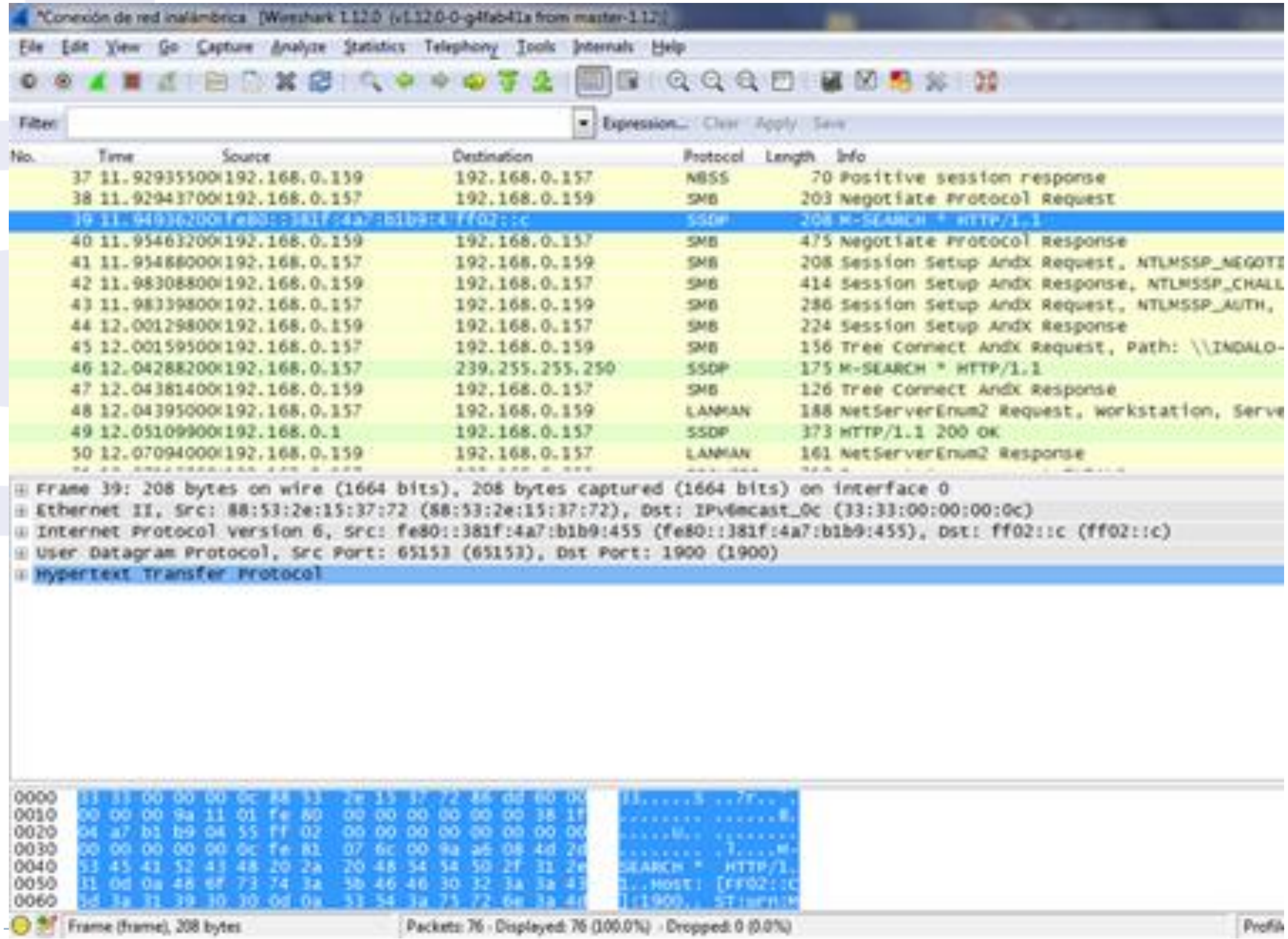
- 1 Introduction to Wireshark
- 2 Capturing IPv6 Traffic
- 3 Capturing 6Lowpan Traffic

Wireshark (I)

- **Wireshark** is a sniffer, a free and open-source packet analyzer, allows packet traces to be sniffed, captured, and analysed
- We can capture packets in an interface and Wireshark understands the protocols used and shows the information in a friendly way
- Features:
  - Available for Windows, Linux y Mac OS
  - Graphical interface
  - Allows for filtering the packet captures
  - Generates statistics and graphs
  - Lot of protocols supported

Wireshark (II)

- 4 areas: menus and filters, list of captured packets, detailed information about the selected packet, full content of selected pkt in hex and ASCII


*Conexión de red inalámbrica   [Wireshark 1.12.0 (v1.12.0-0-g4fab41a from master-1.12)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save

```
No.    Time         Source                    Destination        Protocol  Length  Info
    37 11.92935500192.168.0.159               192.168.0.157      NBSS          70 Positive session response
    38 11.92943700192.168.0.157               192.168.0.159      SMB          203 Negotiate Protocol Request
    39 11.94936200fe80::381f:4a7:b1b9:4       ff02::c            SSDP         208 M-SEARCH * HTTP/1.1
    40 11.95463200192.168.0.159               192.168.0.157      SMB          475 Negotiate Protocol Response
    41 11.95488000192.168.0.157               192.168.0.159      SMB          208 Session Setup AndX Request, NTLMSSP_NEGOTI
    42 11.98308800192.168.0.159               192.168.0.157      SMB          414 Session Setup AndX Response, NTLMSSP_CHALL
    43 11.98339800192.168.0.157               192.168.0.159      SMB          286 Session Setup AndX Request, NTLMSSP_AUTH,
    44 12.00129800192.168.0.159               192.168.0.157      SMB          224 Session Setup AndX Response
    45 12.00159500192.168.0.157               192.168.0.159      SMB          156 Tree Connect AndX Request, Path: \\INDALO-
    46 12.04288200192.168.0.157               239.255.255.250    SSDP         175 M-SEARCH * HTTP/1.1
    47 12.04381400192.168.0.159               192.168.0.157      SMB          126 Tree Connect AndX Response
    48 12.04395000192.168.0.157               192.168.0.159      LANMAN       188 NetServerEnum2 Request, Workstation, Serve
    49 12.05109900192.168.0.1                 192.168.0.157      SSDP         373 HTTP/1.1 200 OK
    50 12.07094000192.168.0.159               192.168.0.157      LANMAN       161 NetServerEnum2 Response
```

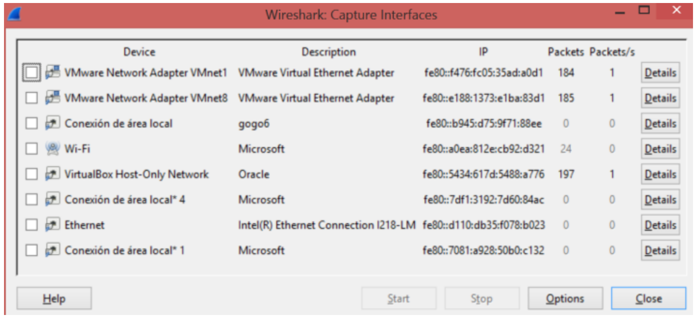
```
⊞ Frame 39: 208 bytes on wire (1664 bits), 208 bytes captured (1664 bits) on interface 0
⊞ Ethernet II, Src: 88:53:2e:15:37:72 (88:53:2e:15:37:72), Dst: IPv6mcast_0c (33:33:00:00:00:0c)
⊞ Internet Protocol Version 6, Src: fe80::381f:4a7:b1b9:455 (fe80::381f:4a7:b1b9:455), Dst: ff02::c (ff02::c)
⊞ User Datagram Protocol, Src Port: 65153 (65153), Dst Port: 1900 (1900)
⊞ Hypertext Transfer Protocol
```

```
0000  33 33 00 00 00 0c 88 53  2e 15 37 72 86 dd 60 00   33.....S ..7r..`.
0010  00 00 00 9a 11 01 fe 80  00 00 00 00 00 00 38 1f   ........ ......8.
0020  04 a7 b1 b9 04 55 ff 02  00 00 00 00 00 00 00 00   .....U.. ........
0030  00 00 00 00 00 0c fe 81  07 6c 00 9a a6 08 4d 2d   ........ .l....M-
0040  53 45 41 52 43 48 20 2a  20 48 54 54 50 2f 31 2e   SEARCH *  HTTP/1.
0050  31 0d 0a 48 6f 73 74 3a  5b 46 46 30 32 3a 3a 43   1..Host: [FF02::C
0060  5d 3a 31 39 30 30 0d 0a  53 54 3a 75 72 6e 3a 4d   ]:1900.. ST:urn:M
```

Frame (frame), 208 bytes | Packets: 76 · Displayed: 76 (100.0%) · Dropped: 0 (0.0%) | Profile

Wireshark (III)

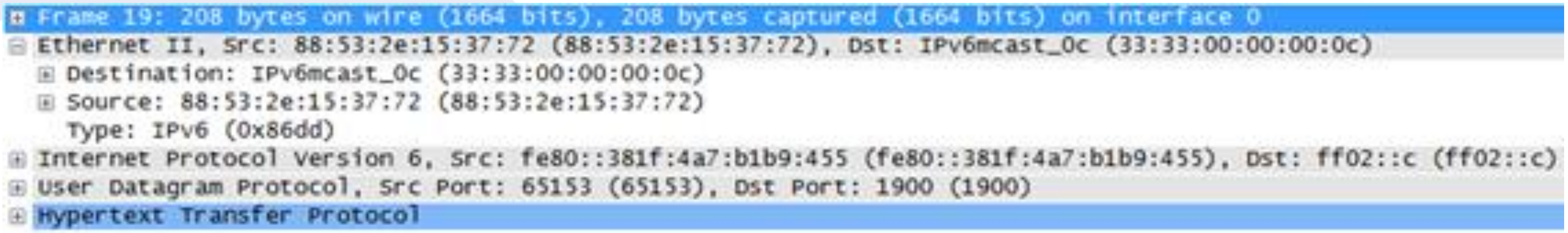
- Files -> Open
  - To open saved capture files
- Help -> Sample Captures
  - Allow to fetch caputre examples
- Capture -> Interfaces…
  - Choos interface(s) in which capture
- Capture -> Options…
  - Configure capture details
- Edit -> Find Packet
  - To look for specific packets


Wireshark: Capture Interfaces

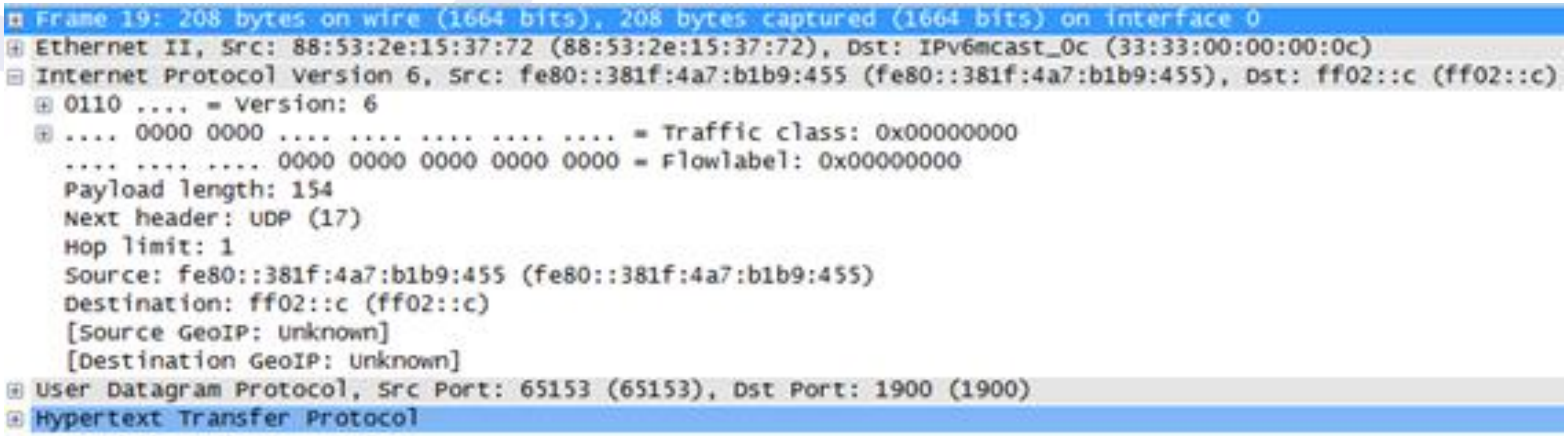
	Device	Description	IP	Packets	Packets/s	
☐	VMware Network Adapter VMnet1	VMware Virtual Ethernet Adapter	fe80::f476:fc05:35ad:a0d1	184	1	Details
☐	VMware Network Adapter VMnet8	VMware Virtual Ethernet Adapter	fe80::e188:1373:e1ba:83d1	185	1	Details
☐	Conexión de área local	gogo6	fe80::b945:d75:9f71:88ee	0	0	Details
☐	Wi-Fi	Microsoft	fe80::a0ea:812e:cb92:d321	24	0	Details
☐	VirtualBox Host-Only Network	Oracle	fe80::5434:617d:5488:a776	197	1	Details
☐	Conexión de área local* 4	Microsoft	fe80::7df1:3192:7d60:84ac	0	0	Details
☐	Ethernet	Intel(R) Ethernet Connection I218-LM	fe80::d110:db35:f078:b023	0	0	Details
☐	Conexión de área local* 1	Microsoft	fe80::7081:a928:50b0:c132	0	0	Details

Wireshark (IV)

- Detailed packet information:
  - Information shown by layers
  - Expand/compress details



```
⊞ Frame 19: 208 bytes on wire (1664 bits), 208 bytes captured (1664 bits) on interface 0
⊟ Ethernet II, Src: 88:53:2e:15:37:72 (88:53:2e:15:37:72), Dst: IPv6mcast_0c (33:33:00:00:00:0c)
  ⊞ Destination: IPv6mcast_0c (33:33:00:00:00:0c)
  ⊞ Source: 88:53:2e:15:37:72 (88:53:2e:15:37:72)
    Type: IPv6 (0x86dd)
⊞ Internet Protocol Version 6, Src: fe80::381f:4a7:b1b9:455 (fe80::381f:4a7:b1b9:455), Dst: ff02::c (ff02::c)
⊞ User Datagram Protocol, Src Port: 65153 (65153), Dst Port: 1900 (1900)
⊞ Hypertext Transfer Protocol
```

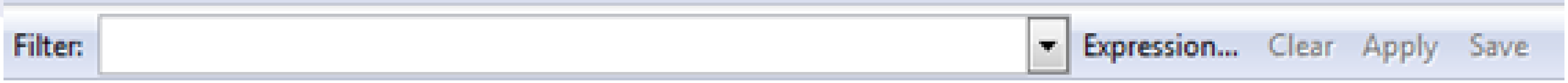


```
⊞ Frame 19: 208 bytes on wire (1664 bits), 208 bytes captured (1664 bits) on interface 0
⊞ Ethernet II, Src: 88:53:2e:15:37:72 (88:53:2e:15:37:72), Dst: IPv6mcast_0c (33:33:00:00:00:0c)
⊟ Internet Protocol Version 6, Src: fe80::381f:4a7:b1b9:455 (fe80::381f:4a7:b1b9:455), Dst: ff02::c (ff02::c)
  ⊞ 0110 .... = Version: 6
  ⊞ .... 0000 0000 .... .... .... .... .... = Traffic class: 0x00000000
    .... .... .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
    Payload length: 154
    Next header: UDP (17)
    Hop limit: 1
    Source: fe80::381f:4a7:b1b9:455 (fe80::381f:4a7:b1b9:455)
    Destination: ff02::c (ff02::c)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
⊞ User Datagram Protocol, Src Port: 65153 (65153), Dst Port: 1900 (1900)
⊞ Hypertext Transfer Protocol
```

Wireshark (V)

- Two ways of applying **Filters**:

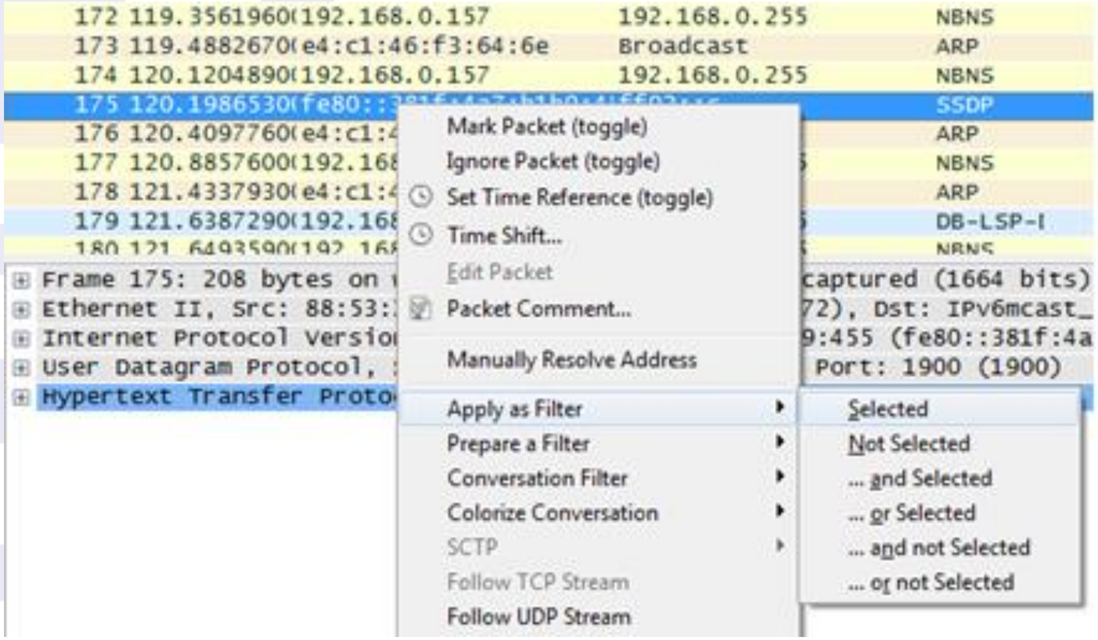
1. Write filter expression and apply it.
    - Protocols (ip, ipv6, icmp, icmpv6)
    - Protocol field (ipv6.dst, ipv6.src)
    - Complex expressions using operators: AND (&&), OR (II) or negation (I)



Wireshark (VI)

- Two ways of applying **Filters**(cont.):

2. Right click in one filed of a captured packet
   - In the packet list
   - Appear a menu option "Apply as filter“ with several options

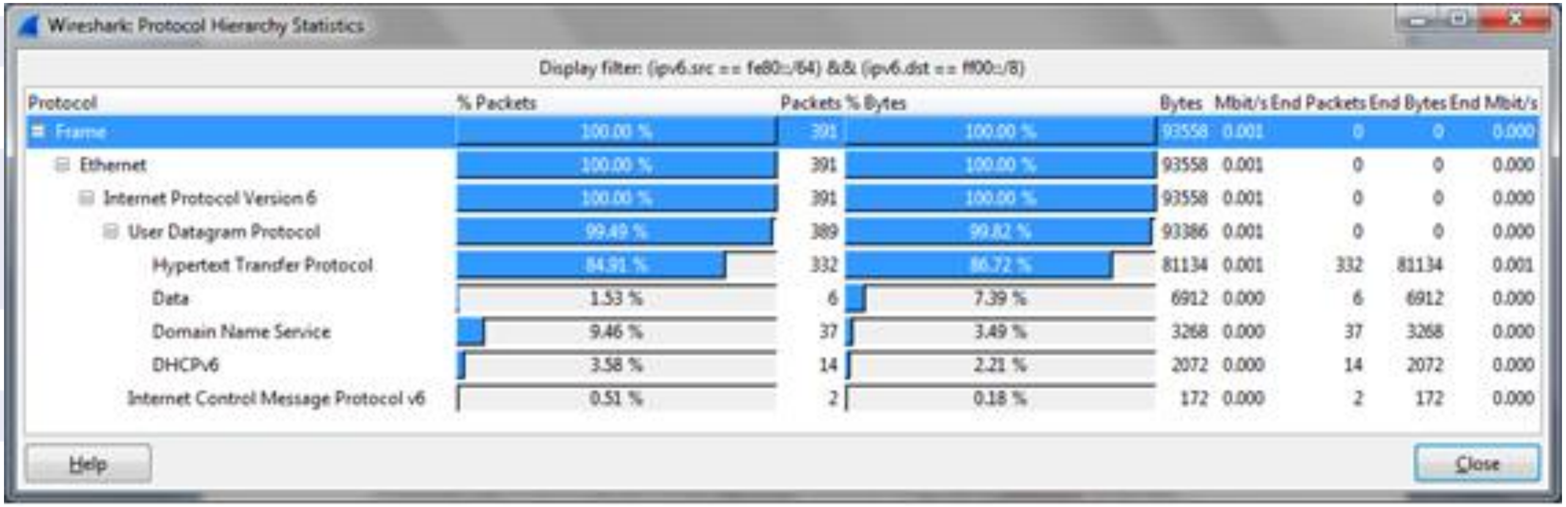


```
172 119.3561960(192.168.0.157       192.168.0.255       NBNS
173 119.4882670(e4:c1:46:f3:64:6e   Broadcast           ARP
174 120.1204890(192.168.0.157       192.168.0.255       NBNS
175 120.1986530(fe80::                                  SSDP
176 120.4097760(e4:c1:4                                 ARP
177 120.8857600(192.168                              5  NBNS
178 121.4337930(e4:c1:4                                 ARP
179 121.6387290(192.168                              5  DB-LSP-I
180 121.6493590(192.168                              5  NBNS
```

```
⊞ Frame 175: 208 bytes on                captured (1664 bits)
⊞ Ethernet II, Src: 88:53:               72), Dst: IPv6mcast_
⊞ Internet Protocol Versio               9:455 (fe80::381f:4a
⊞ User Datagram Protocol,                 Port: 1900 (1900)
⊞ Hypertext Transfer Proto
```

Wireshark (VII)

- **statistics about the captured traffic**:
  - With applied filters, the statistics will be about the filtered traffic
  - **Statistics** and select, for example, **Protocol Hierarchy**



Wireshark: Protocol Hierarchy Statistics

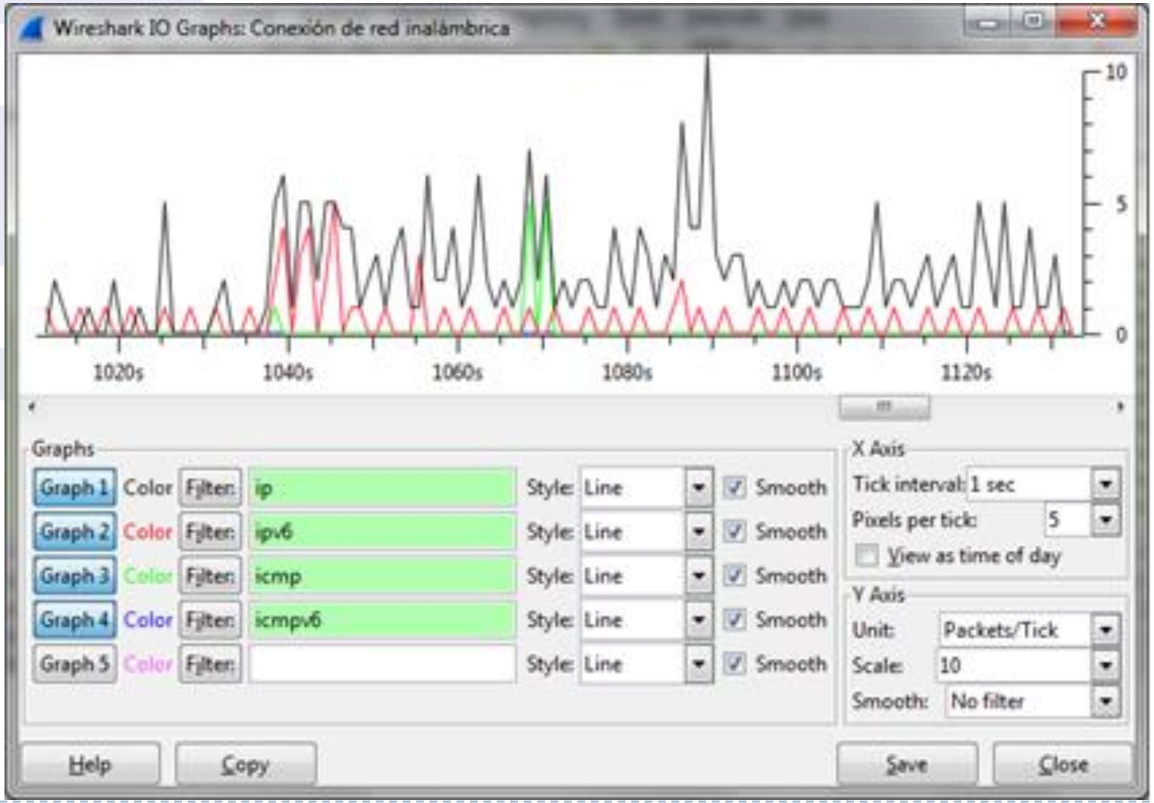
Display filter: (ipv6.src == fe80::/64) && (ipv6.dst == ff00::/8)

Protocol	% Packets	Packets	% Bytes	Bytes	Mbit/s	End Packets	End Bytes	End Mbit/s
⊟ Frame	100.00 %	391	100.00 %	93558	0.001	0	0	0.000
⊟ Ethernet	100.00 %	391	100.00 %	93558	0.001	0	0	0.000
⊟ Internet Protocol Version 6	100.00 %	391	100.00 %	93558	0.001	0	0	0.000
⊟ User Datagram Protocol	99.49 %	389	99.82 %	93386	0.001	0	0	0.000
Hypertext Transfer Protocol	84.91 %	332	86.72 %	81134	0.001	332	81134	0.001
Data	1.53 %	6	7.39 %	6912	0.000	6	6912	0.000
Domain Name Service	9.46 %	37	3.49 %	3268	0.000	37	3268	0.000
DHCPv6	3.58 %	14	2.21 %	2072	0.000	14	2072	0.000
Internet Control Message Protocol v6	0.51 %	2	0.18 %	172	0.000	2	172	0.000

- Other interesting options are:
  - Conversation List ---> IPv6
  - Statistics ---> Endpoint List ---> IPv6
  - Statistics ---> IO Graph

Wireshark (VIII)

- Statistics ---> IO Graph
  - Allow to create and save graphs
  - Different lines for different types of traffic (filters)



Wireshark IO Graphs: Conexión de red inalámbrica

Graphs

Graph 1 Color Filter: ip Style: Line ☑ Smooth

Graph 2 Color Filter: ipv6 Style: Line ☑ Smooth

Graph 3 Color Filter: icmp Style: Line ☑ Smooth

Graph 4 Color Filter: icmpv6 Style: Line ☑ Smooth

Graph 5 Color Filter: Style: Line ☑ Smooth

X Axis

Tick interval: 1 sec

Pixels per tick: 5

□ View as time of day

Y Axis

Unit: Packets/Tick

Scale: 10

Smooth: No filter

Wireshark: Exercises (I)

- Exercise A: Capture packets on eth0 interface in your RPi
  - Filter by protocols: IPv4, IPv6, ICMPv6
  - Look into protocol details of Ethernet, IPv4/IPv6, etc.
- Exercise B: Apply Filters
  - Show only IPv6 traffic
  - Only ICMPv6
  - Show pkts with your link-local address as source
  - Show pkts with your link-local address as source AND destination
  - Show only ICMPv6 type NA and NS

Wireshark: Exercises (II)

- Exercise C: See statistics of captured traffic by protocols

- Exercise D: Generate a graph showing different lines for IPv4, IPv6 and ICMPv6

Capturing 6Lowpan Traffic (I)

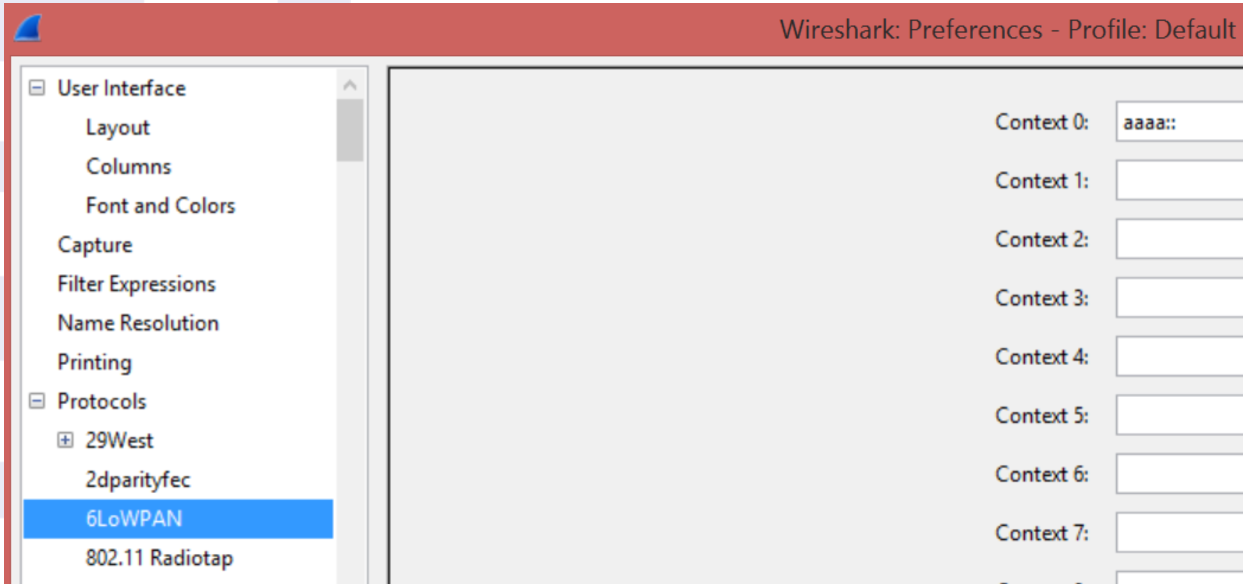
- Live demo of 6Lowpan capturing packets

Capturing 6Lowpan Traffic (II)

- Open the capture file: wireshark-ipv6-6lowpan.pcap

- You need to change some things on Wireshark:

1. Edit -> Preferences ->
2. Protocols -> 6lowpan -> context0: aaaa::



Wireshark: Preferences - Profile: Default

Context 0: aaaa::

Context 1:

Context 2:

Context 3:

Context 4:

Context 5:

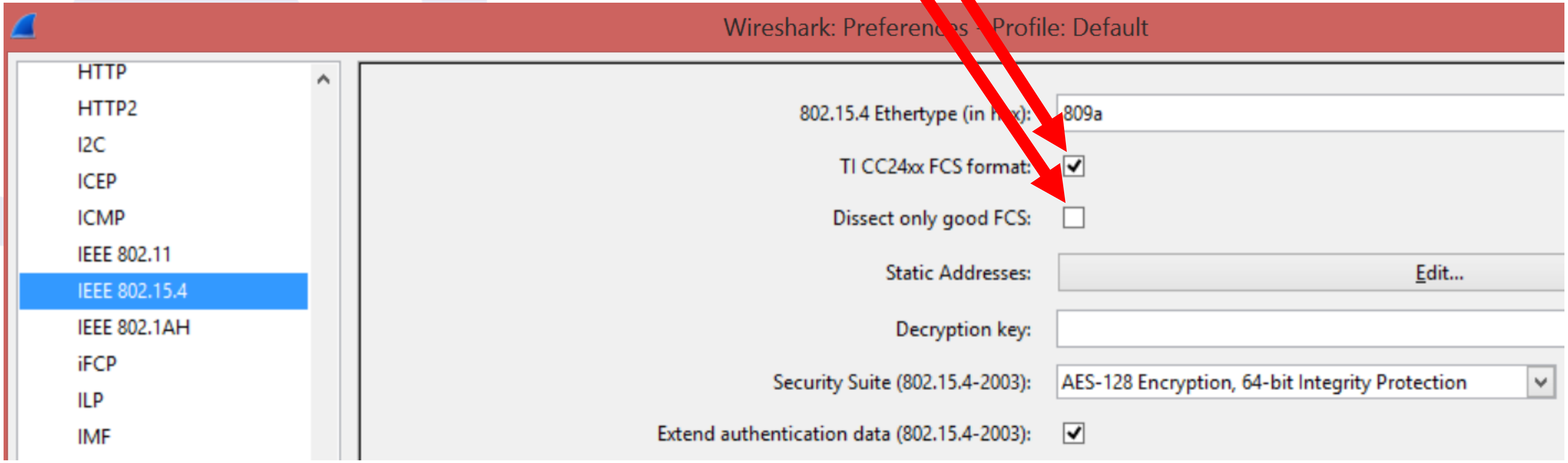
Context 6:

Context 7:

Capturing 6Lowpan Traffic (III)

- You need to change some things on Wireshark (cont.):

1. Edit -> Preferences ->
2. Protocols -> IEEE 802.15.4



Wireshark: Preferences - Profile: Default

802.15.4 Ethertype (in hex): 809a

TI CC24xx FCS format: ☑

Dissect only good FCS: □

Static Addresses: Edit...

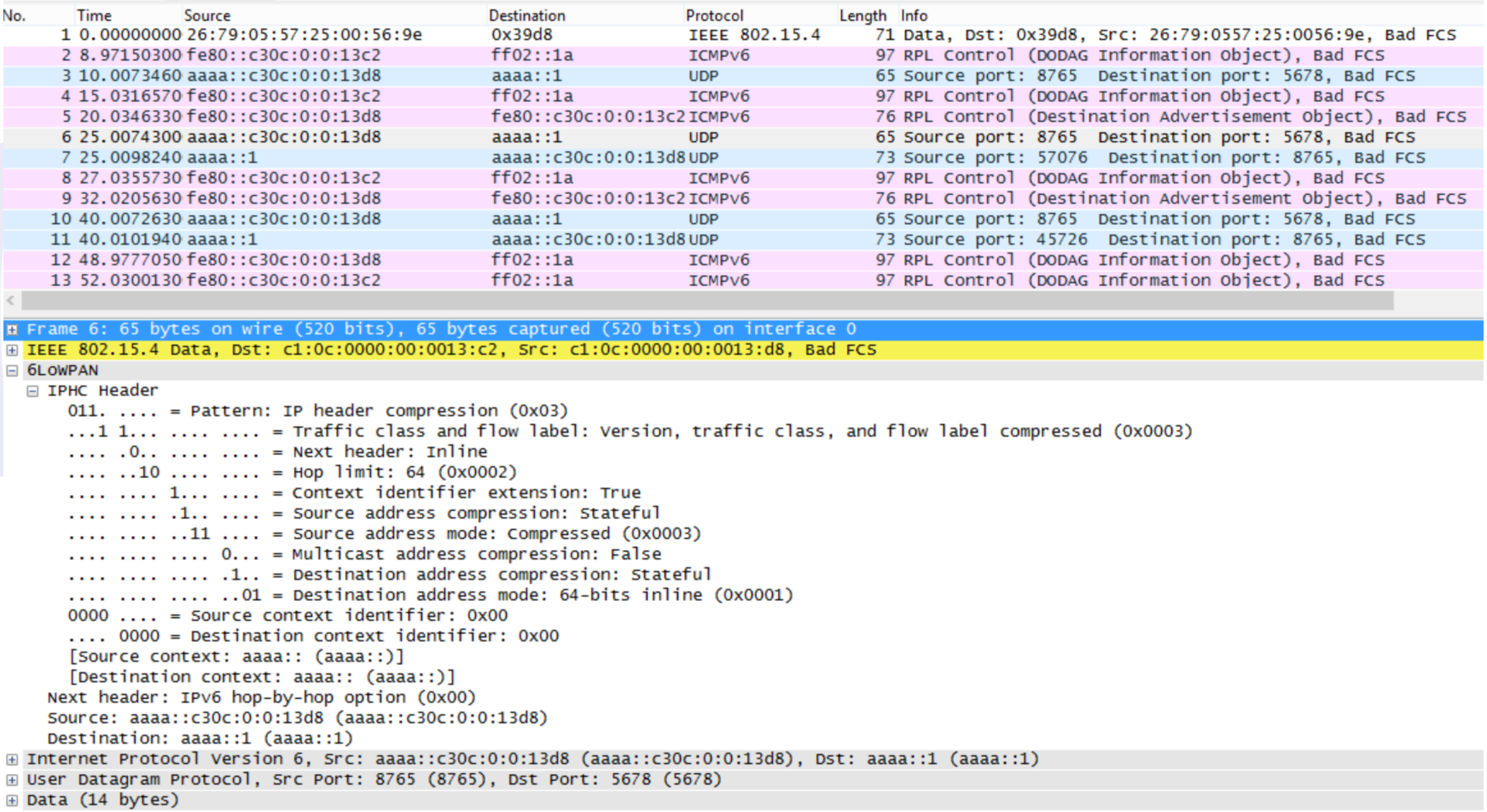
Decryption key:

Security Suite (802.15.4-2003): AES-128 Encryption, 64-bit Integrity Protection

Extend authentication data (802.15.4-2003): ☑

Capturing 6Lowpan Traffic (IV)

- You can see information of the different layers



No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000000	26:79:05:57:25:00:56:9e	0x39d8	IEEE 802.15.4	71	Data, Dst: 0x39d8, Src: 26:79:0557:25:0056:9e, Bad FCS
2	8.97150300	fe80::c30c:0:0:13c2	ff02::1a	ICMPv6	97	RPL Control (DODAG Information Object), Bad FCS
3	10.0073460	aaaa::c30c:0:0:13d8	aaaa::1	UDP	65	Source port: 8765 Destination port: 5678, Bad FCS
4	15.0316570	fe80::c30c:0:0:13c2	ff02::1a	ICMPv6	97	RPL Control (DODAG Information Object), Bad FCS
5	20.0346330	fe80::c30c:0:0:13d8	fe80::c30c:0:0:13c2	ICMPv6	76	RPL Control (Destination Advertisement Object), Bad FCS
6	25.0074300	aaaa::c30c:0:0:13d8	aaaa::1	UDP	65	Source port: 8765 Destination port: 5678, Bad FCS
7	25.0098240	aaaa::1	aaaa::c30c:0:0:13d8	UDP	73	Source port: 57076 Destination port: 8765, Bad FCS
8	27.0355730	fe80::c30c:0:0:13c2	ff02::1a	ICMPv6	97	RPL Control (DODAG Information Object), Bad FCS
9	32.0205630	fe80::c30c:0:0:13d8	fe80::c30c:0:0:13c2	ICMPv6	76	RPL Control (Destination Advertisement Object), Bad FCS
10	40.0072630	aaaa::c30c:0:0:13d8	aaaa::1	UDP	65	Source port: 8765 Destination port: 5678, Bad FCS
11	40.0101940	aaaa::1	aaaa::c30c:0:0:13d8	UDP	73	Source port: 45726 Destination port: 8765, Bad FCS
12	48.9777050	fe80::c30c:0:0:13d8	ff02::1a	ICMPv6	97	RPL Control (DODAG Information Object), Bad FCS
13	52.0300130	fe80::c30c:0:0:13c2	ff02::1a	ICMPv6	97	RPL Control (DODAG Information Object), Bad FCS

```
⊞ Frame 6: 65 bytes on wire (520 bits), 65 bytes captured (520 bits) on interface 0
⊞ IEEE 802.15.4 Data, Dst: c1:0c:0000:00:0013:c2, Src: c1:0c:0000:00:0013:d8, Bad FCS
⊟ 6LoWPAN
  ⊟ IPHC Header
      011. .... = Pattern: IP header compression (0x03)
      ...1 1... .... .... = Traffic class and flow label: Version, traffic class, and flow label compressed (0x0003)
      .... .0.. .... .... = Next header: Inline
      .... ..10 .... .... = Hop limit: 64 (0x0002)
      .... .... 1... .... = Context identifier extension: True
      .... .... .1.. .... = Source address compression: Stateful
      .... .... ..11 .... = Source address mode: Compressed (0x0003)
      .... .... .... 0... = Multicast address compression: False
      .... .... .... .1.. = Destination address compression: Stateful
      .... .... .... ..01 = Destination address mode: 64-bits inline (0x0001)
      0000 .... = Source context identifier: 0x00
      .... 0000 = Destination context identifier: 0x00
      [Source context: aaaa:: (aaaa::)]
      [Destination context: aaaa:: (aaaa::)]
    Next header: IPv6 hop-by-hop option (0x00)
    Source: aaaa::c30c:0:0:13d8 (aaaa::c30c:0:0:13d8)
    Destination: aaaa::1 (aaaa::1)
⊞ Internet Protocol Version 6, Src: aaaa::c30c:0:0:13d8 (aaaa::c30c:0:0:13d8), Dst: aaaa::1 (aaaa::1)
⊞ User Datagram Protocol, Src Port: 8765 (8765), Dst Port: 5678 (5678)
⊞ Data (14 bytes)
```

Thanks!

Questions?

- Contact: info@nodo6.com / training@nodo6.com
- http://www.nodo6.com
- https://www.linkedin.com/company/nodo6
- https://twitter.com/NODO6_RRSS