POLITECNICO
MILANO 1863

Sniffing and processing wireless traffic

[redacted]
Karo Amini

[redacted]

Sniffing

- **Sniffing** or **eavesdropping** is the process of secretly listening to the communication of others (even without their consent)

- For wireless networks, sniffing can be performed just by tuning a receiver on the correct transmission frequency and by knowing what communication protocol is used

- Clearly, most of the time the original communication is encrypted so that only who has the right ‘key’ (WPA, WPA2, for Wi-Fi, KASUMI block cipher in 3G/LTE)

Objectives of this lecture

- Learn to sniff WiFi traffic

- Explore the PHY and MAC layers of WiFi and its management functions

- Analyze and process the captured data to answer the following questions:
  - How many WiFi devices are present in this room?
  - What is the most popular vendor?
  - Other?

- Tools we will use:
  - Wireshark (for sniffing and manually analyzing traffic)
  - Python (for automatically analyzing data and visualizing results)

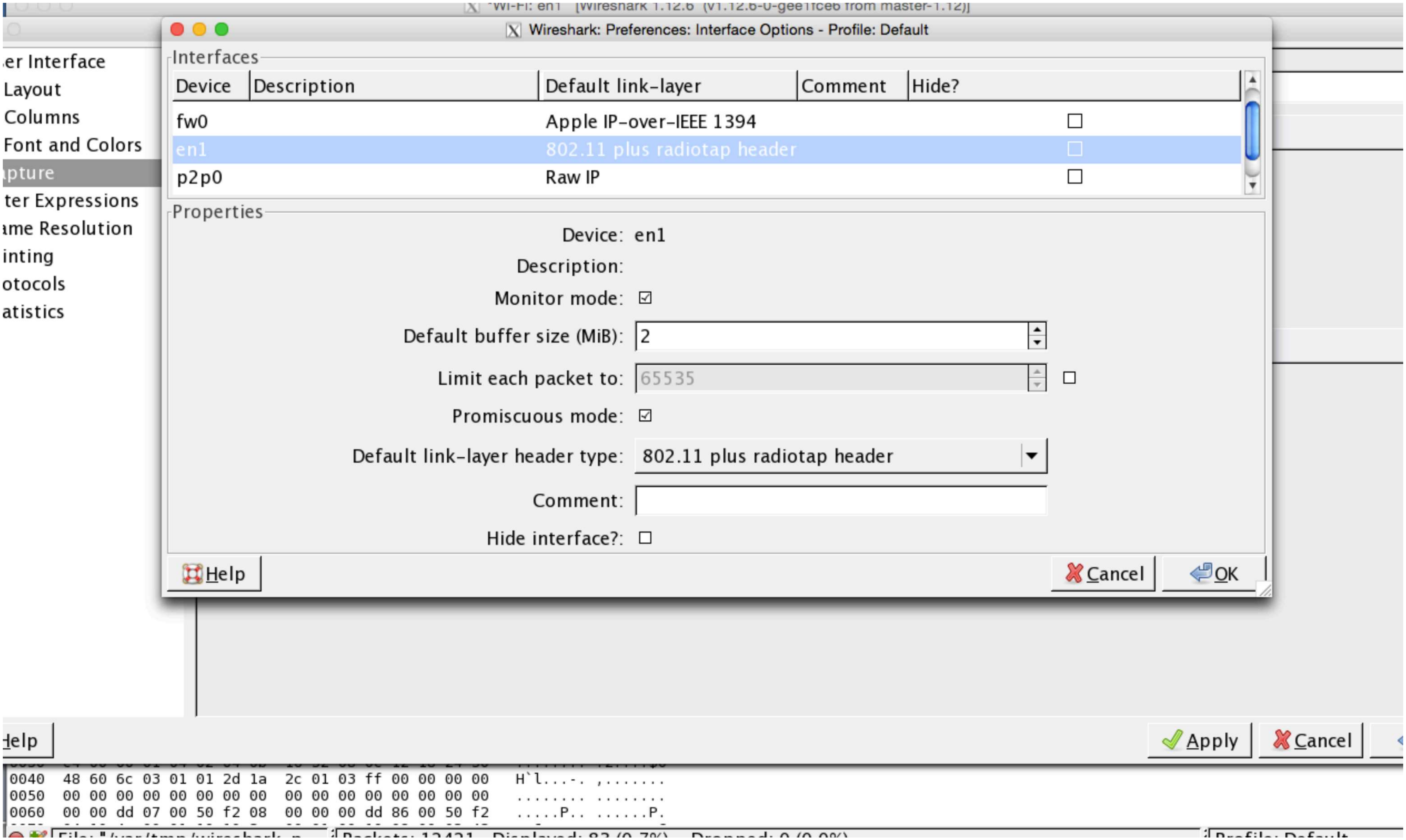
Using the “monitor” mode

- One of the 7 modes 802.11 (most) wireless cards can operate in

- It allows to capture packets on a particular Wi-Fi channel without the need of being associated with a network first.

- To activate monitor mode, administrator rights are needed:
  - Linux (interface “wlan0” on channel 6)
    - sudo ifconfig wlan0 down/up
    - sudo iwconfig wlan0 mode monitor chan 6
  - Mac OS X (interface “en1” on channel 6)
    - sudo airport en1 sniff 6
  - Windows
    - Specific software (e.g. Acrylic / Microsoft Network Monitor)
  - In general, it can be done directly from Wireshark (if executed with administrator rights)

Monitor mode in wireshark



Device	Description	Default link-layer	Comment	Hide?
fw0		Apple IP-over-IEEE 1394		□
en1		802.11 plus radiotap header		□
p2p0		Raw IP		□

Device: en1

Description:

Monitor mode: ☑

Default buffer size (MiB): 2

Limit each packet to: 65535 □

Promiscuous mode: ☑

Default link-layer header type: 802.11 plus radiotap header

Comment:

Hide interface?: □

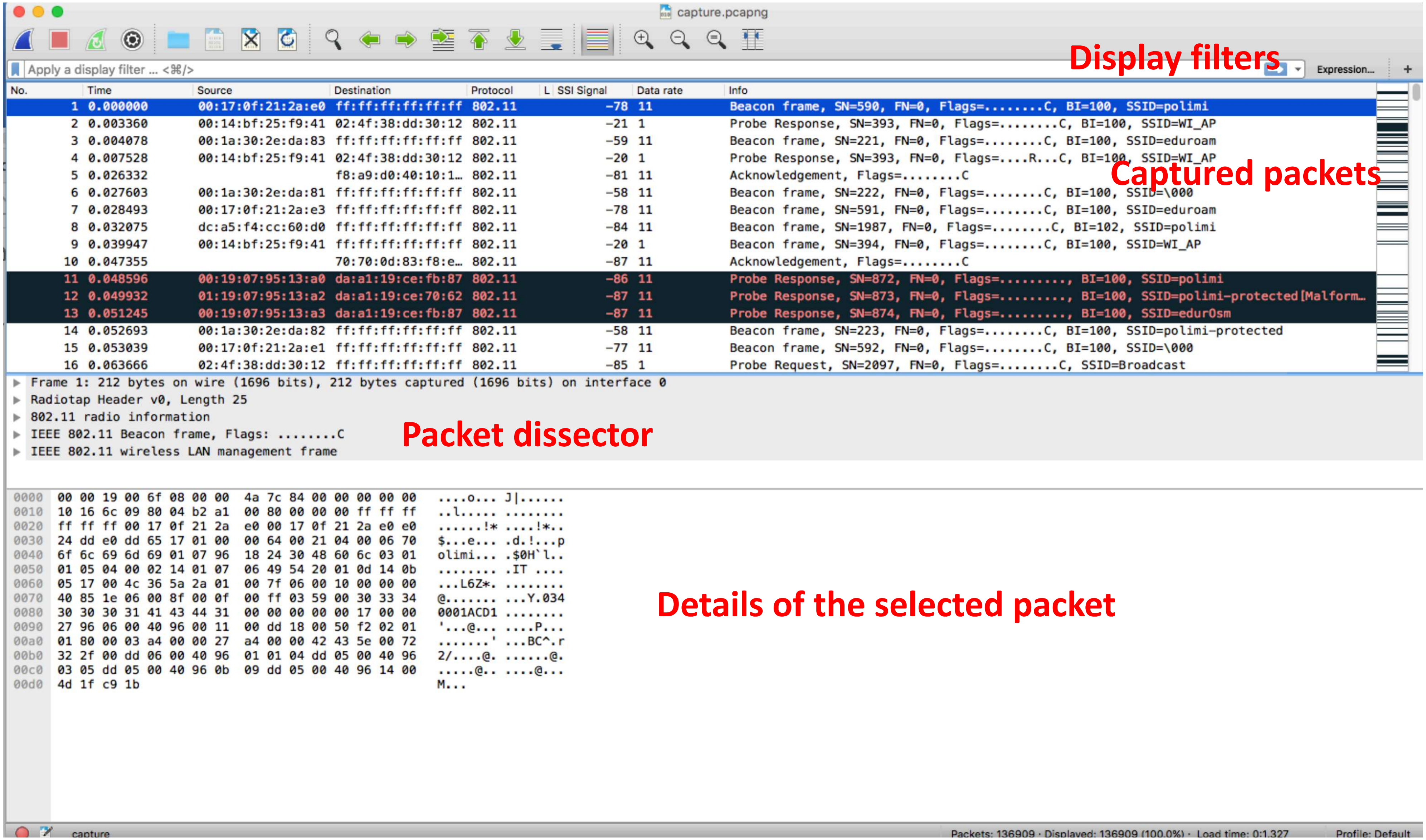
```
0040  48 60 6c 03 01 01 2d 1a  2c 01 03 ff 00 00 00 00   H`l...-. ,.......
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0060  00 00 dd 07 00 50 f2 08  00 00 00 dd 86 00 50 f2   .....P.. ......P.
```

File: "/var/tmp/wireshark_p... Packets: 12421 · Displayed: 83 (0.7%) · Dropped: 0 (0.0%) Profile: Default

Wireshark

- Gold standard open source software for capturing and analyzing network traffic
  - Generally used to inspect / solve network issues
  - Based on a graphical user interface
  - Already contains many protocol *dissectors*
- Let’s play with it
  - Open Wireshark
  - Load the ”office_capture.pcapng” file available on the website. It contains about 1 minute of Wi-Fi traffic captured in monitor mode in my office
  - Let’s learn how to use the software…

Wireshark main window



Display filters

Captured packets

```
No.   Time      Source             Destination        Protocol  L SSI Signal  Data rate  Info
    1 0.000000  00:17:0f:21:2a:e0  ff:ff:ff:ff:ff:ff  802.11          -78  11        Beacon frame, SN=590, FN=0, Flags=........C, BI=100, SSID=polimi
    2 0.003360  00:14:bf:25:f9:41  02:4f:38:dd:30:12  802.11          -21  1         Probe Response, SN=393, FN=0, Flags=........C, BI=100, SSID=WI_AP
    3 0.004078  00:1a:30:2e:da:83  ff:ff:ff:ff:ff:ff  802.11          -59  11        Beacon frame, SN=221, FN=0, Flags=........C, BI=100, SSID=eduroam
    4 0.007528  00:14:bf:25:f9:41  02:4f:38:dd:30:12  802.11          -20  1         Probe Response, SN=393, FN=0, Flags=....R...C, BI=100, SSID=WI_AP
    5 0.026332                     f8:a9:d0:40:10:1…  802.11          -81  11        Acknowledgement, Flags=........C
    6 0.027603  00:1a:30:2e:da:81  ff:ff:ff:ff:ff:ff  802.11          -58  11        Beacon frame, SN=222, FN=0, Flags=........C, BI=100, SSID=\000
    7 0.028493  00:17:0f:21:2a:e3  ff:ff:ff:ff:ff:ff  802.11          -78  11        Beacon frame, SN=591, FN=0, Flags=........C, BI=100, SSID=eduroam
    8 0.032075  dc:a5:f4:cc:60:d0  ff:ff:ff:ff:ff:ff  802.11          -84  11        Beacon frame, SN=1987, FN=0, Flags=........C, BI=102, SSID=polimi
    9 0.039947  00:14:bf:25:f9:41  ff:ff:ff:ff:ff:ff  802.11          -20  1         Beacon frame, SN=394, FN=0, Flags=........C, BI=100, SSID=WI_AP
   10 0.047355                     70:70:0d:83:f8:e…  802.11          -87  11        Acknowledgement, Flags=........C
   11 0.048596  00:19:07:95:13:a0  da:a1:19:ce:fb:87  802.11          -86  11        Probe Response, SN=872, FN=0, Flags=.........,  BI=100, SSID=polimi
   12 0.049932  01:19:07:95:13:a2  da:a1:19:ce:70:62  802.11          -87  11        Probe Response, SN=873, FN=0, Flags=.........,  BI=100, SSID=polimi-protected[Malform…
   13 0.051245  00:19:07:95:13:a3  da:a1:19:ce:fb:87  802.11          -87  11        Probe Response, SN=874, FN=0, Flags=.........,  BI=100, SSID=edur0sm
   14 0.052693  00:1a:30:2e:da:82  ff:ff:ff:ff:ff:ff  802.11          -58  11        Beacon frame, SN=223, FN=0, Flags=........C, BI=100, SSID=polimi-protected
   15 0.053039  00:17:0f:21:2a:e1  ff:ff:ff:ff:ff:ff  802.11          -77  11        Beacon frame, SN=592, FN=0, Flags=........C, BI=100, SSID=\000
   16 0.063666  02:4f:38:dd:30:12  ff:ff:ff:ff:ff:ff  802.11          -85  1         Probe Request, SN=2097, FN=0, Flags=........C, SSID=Broadcast
```

```
▶ Frame 1: 212 bytes on wire (1696 bits), 212 bytes captured (1696 bits) on interface 0
▶ Radiotap Header v0, Length 25
▶ 802.11 radio information
▶ IEEE 802.11 Beacon frame, Flags: ........C
▶ IEEE 802.11 wireless LAN management frame
```

Packet dissector

```
0000  00 00 19 00 6f 08 00 00  4a 7c 84 00 00 00 00 00   ....o... J|......
0010  10 16 6c 09 80 04 b2 a1  00 80 00 00 00 ff ff ff   ..l..... ........
0020  ff ff ff 00 17 0f 21 2a  e0 00 17 0f 21 2a e0 e0   ......!* ....!*..
0030  24 dd e0 dd 65 17 01 00  00 64 00 21 04 00 06 70   $...e... .d.!...p
0040  6f 6c 69 6d 69 01 07 96  18 24 30 48 60 6c 03 01   olimi... .$0H`l..
0050  01 05 04 00 02 14 01 07  06 49 54 20 01 0d 14 0b   ........ .IT ....
0060  05 17 00 4c 36 5a 2a 01  00 7f 06 00 10 00 00 00   ...L6Z*. ........
0070  40 85 1e 06 00 8f 00 0f  00 ff 03 59 00 30 33 34   @....... ...Y.034
0080  30 30 30 31 41 43 44 31  00 00 00 00 00 17 00 00   0001ACD1 ........
0090  27 96 06 00 40 96 00 11  00 dd 18 00 50 f2 02 01   '...@... ....P...
00a0  01 80 00 03 a4 00 00 27  a4 00 00 42 43 5e 00 72   .......' ...BC^.r
00b0  32 2f 00 dd 06 00 40 96  01 01 04 dd 05 00 40 96   2/....@. ......@.
00c0  03 05 dd 05 00 40 96 0b  09 dd 05 00 40 96 14 00   .....@.. ....@...
00d0  4d 1f c9 1b                                        M...
```

Details of the selected packet

```
capture                                   Packets: 136909 · Displayed: 136909 (100.0%) · Load time: 0:1.327   Profile: Default
```

Beacon frames

- The first packet is a beacon frame
- Inspect the Radiotap header and 802.11 radio information.
  - Such information are not carried by the packet, they are just added by wireshark when the packet is captured.
  - Interesting ones are Data Rate, Channel Frequency, SSI
- Inspect the 802.11 MAC header
  - Type/subtype, FCF, flags, duration, addresses, etc…

- How often beacons of this network are transmitted?

Data frames

- Find a **data** frame (e.g., frame no. 206)
- Inspect the Radiotap header and 802.11 radio information.
  - Such information are not carried by the packet, they are just added by wireshark when the packet is captured.
  - Interesting ones are Data Rate, Channel Frequency, SSI
- Inspect the 802.11 MAC header
  - Type/subtype, FCF, flags, duration, addresses, etc…

- Create a filter to display only data frames transmitted or received by my smartphone:

```
 (wlan.sa == 44:78:3e:a8:57:a1 or wlan.da == 44:78:3e:a8:57:a1) and
wlan.fc.type_subtype==0x0028
```

ACK frames

- Inspect the first data packet sent by my smartphone (e.g., no 1545)
- What is the type of the following packet (no 1546)?
  - What is its length, compared to the data?
  - Which addresses are contained?
  - Why in your opinion there is no source address?

Retransmitted frames

- We can check the ’Retry’ flag to understand if a frame was retransmitted (corresponding filter: wlan.fc.retry == 1)

- How to count how many data frames sent by my smartphone were retransmitted?

```
wlan.sa == 44:78:3e:a8:57:a1 and wlan.fc.type==2 and
wlan.fc.retry==1
```

- What about received frames? Is the Packet Error Rate simmetrical?

- In this case, it seems that downlink PER is half of uplink PER

Power management

- Remember that the power management bit is set to one when a station is going to sleep

- Is my smartphone going to sleep?

```
wlan.sa == 44:78:3e:a8:57:a1 and wlan.pwrmgt==1
```

- What happens at beacon frame no 3691?

Association

- Let's find out if some device associated while we were sniffing traffic:

  ```
  wlan.fc.type_subtype==0
  ```

- Look at packet no 113488. It's an association request

- Where is the response? What AID does it contain?

Probe requests

- Probe requests are used for performing **active scanning**
- They are transmitted even if the device is not connected to the network

- Let’s search for probe requests in the capture

```
wlan.fc.type_subtype==0x04
```

- Which are the most ‘searched’ SSID?

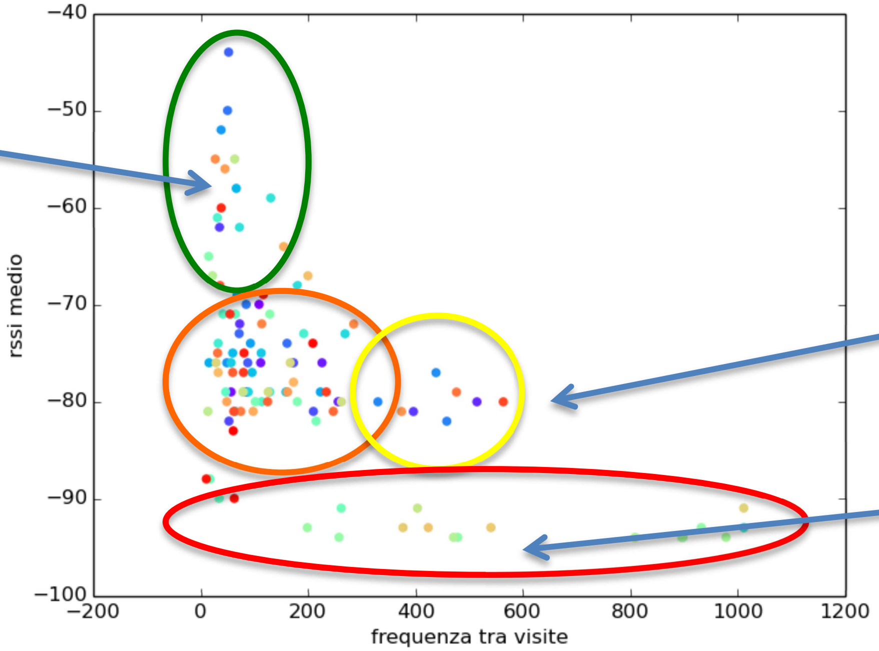
- What information can be inferred from each probe request?

A ‘small’ example in python

- Let’s see what kind of information can be extracted from this room...
  - How many devices are present?
  - How far from the receiver are they?
  - What other information can be extracted?

Other applications

- User behavior estimation
  - How often a user come?
  - Does it stay for a long time?



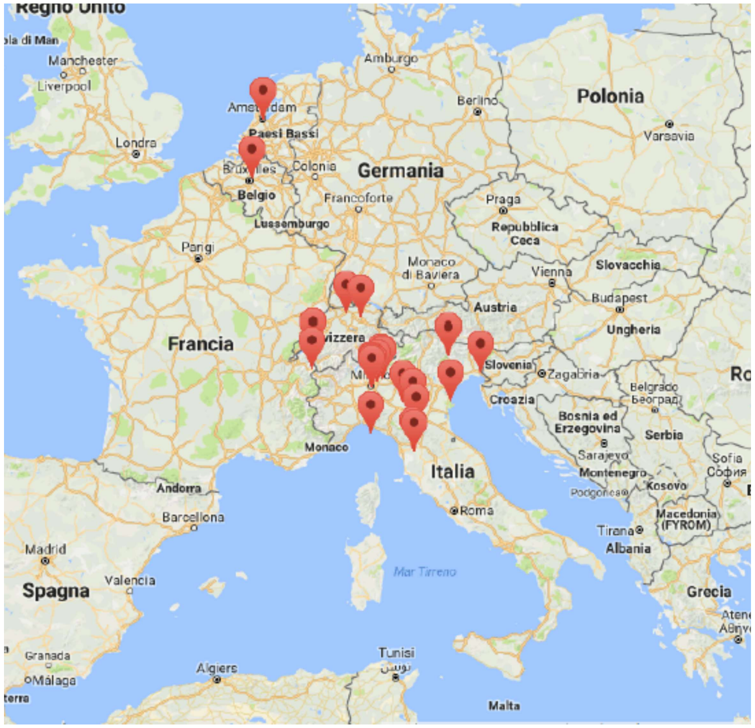
Frequent User

Pass.By User

“Noise”

Using Wigle.net

- A publicly available database to geolocalize SSID…
- What applications can be built on this service?



- Locations of SSID from a 10 minutes scan in a shop near Central station…

Interesting papers

- [1] A. Redondi et al. “Passive Classification of WiFi enabled devices” – MSWIM 2016
- [2] Di Nunzio et al. “Mind Your Probes: De-Anonymization of Large Crowds Through Smartphone WiFi Probe Request” - Infocom 2016
- [3] M. Vanhoef et al. “Why MAC Address Randomization is not Enough: An Analysis of Wi-Fi Network Discovery Mechani” – ASIACCS 2016

Multiple capturing device

- What if the probe requests are captured by more than 1 capturing device?