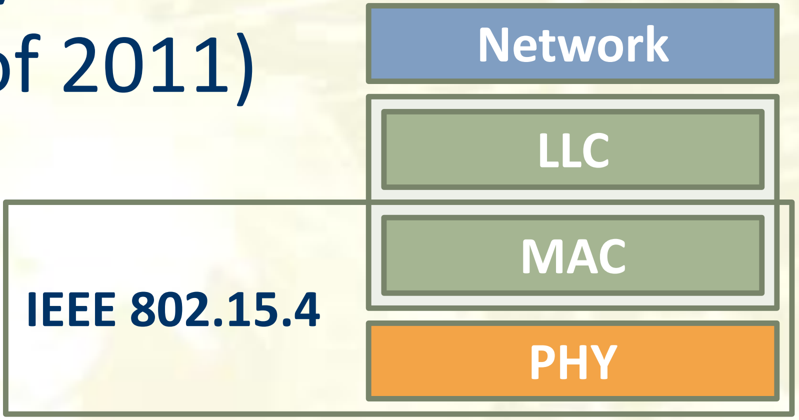
IEEE 802.15.4 Refresher

Carlo Vallati
Assistant Professor @ University of Pisa
c.vallati@iet.unipi.it

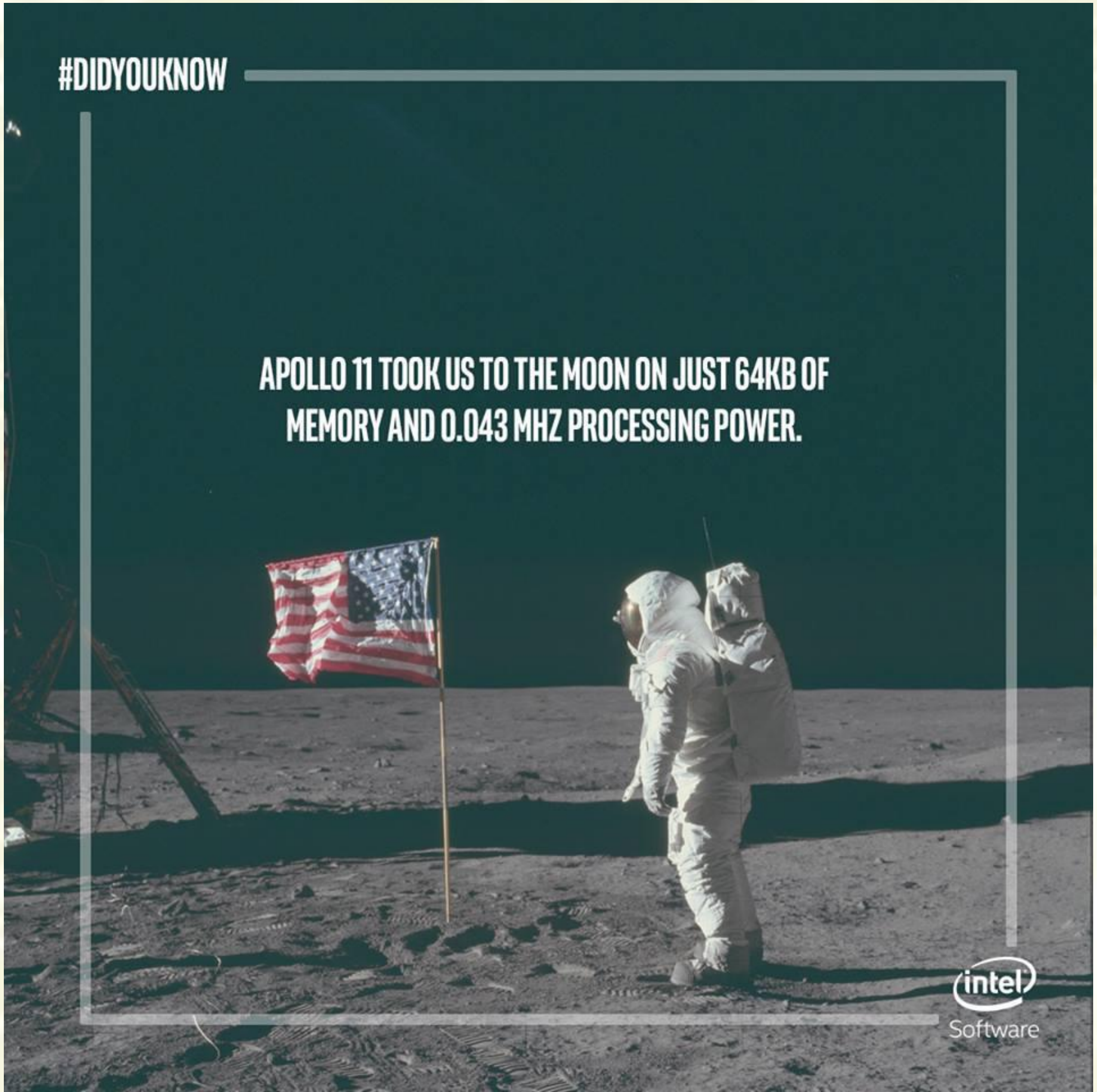
IEEE 802.15.4 standard

- Standard PHY and MAC layers for low-rate WPANs (latest release as of 2011)



- Goal
  - Defining a communication standard for constrained devices with limited computation, power (battery powered devices) and memory

Limited? how much?



IEEE 802.15.4 features

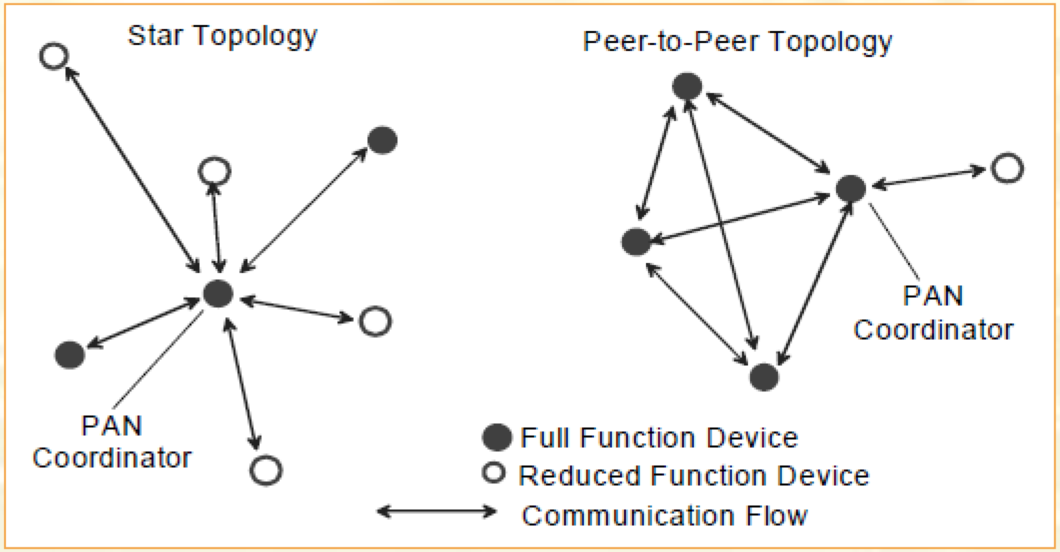
- Main features
  - Low data rate: 20-250 Kbit/s data rates
  - Pure CSMA or hybrid TDMA/CSMA MAC protocols
  - 127 bytes max frame size
  - Long (64-bit) and short (16-bit) addressing modes
  - Star and peer-to-peer network operation
  - Link layer security

IEEE 802.15.4 topologies

- Full vs. Reduced Function Devices
  - FFDs can talk to RFDs or other FFDs, while an RFD can talk only to an FFD
- An RFD is intended for applications that are extremely simple
- The RFD can be implemented using minimal resources and memory capacity
- A full-function device (FFD) has more resources and it is capable of **relaying** messages
- FFDs can operate in three modes: *regular device*, *coordinator* and *PAN coordinator*

IEEE 802.15.4 topologies

- Star vs. P2P topologies
  - **Star**: the communication is established between devices and the single *central controller*, the *PAN coordinator*
  - **P2P**: any device can communicate with any other device in range. *Mesh functionalities for multi-hop data delivery can added at the higher layer, <u>but are not part of this standard</u>*
- PAN unique id
- Coordinators provide synchronization services to other devices



IEEE 802.15.4 addressing

- 64-bit addresses based on IEEE EUI-64, a globally unique id assigned by the manufacturer

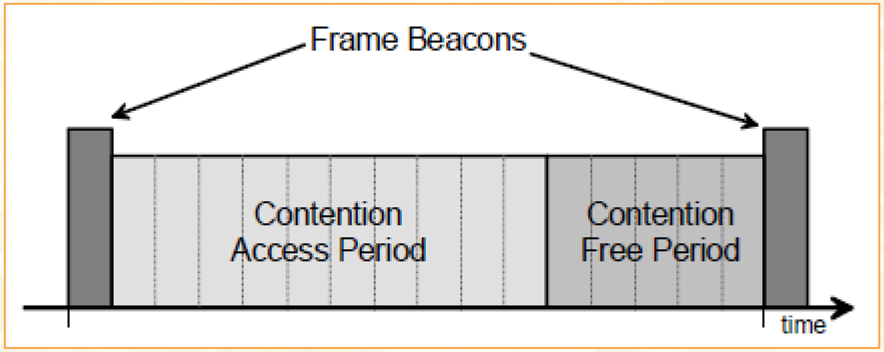
```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     OUI     |L|M|          OUI (cont.)          |               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               +
|                   extension identifier                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

M = multicast
L = local

- Short 16-bit addresses dynamically assigned during network formation
- Source and destination addresses are augmented by the 16-bit *PAN id*

IEEE 802.15.4 operation modes

- Beaconless vs. beacon-enabled MAC operation
- Beaconless mode
  - uses a pure CSMA channel access and operates quite like basic IEEE 802.11
- Beacon-enabled mode
  - superframe structure and the possibility to reserve time-slots for critical data



IEEE 802.15.4 Frame format

- MAC frame format
  - 127 bytes max
  - 88 bytes payload in the worst case

Octets: 2	1	0/2	0/2/8	0/2	0/2/8	0/5/6/10/14	variable	2
Frame Control	Sequence Number	Destination PAN Identifier	Destination Address	Source PAN Identifier	Source Address	Auxiliary Security Header	Frame Payload	FCS
		Addressing fields						
MHR							MAC Payload	MFR

Sniffer

Carlo Vallati
Assistant Professor @ University of Pisa
c.vallati@iet.unipi.it

Sniffer

- Sniffer, what's this thing?

Sniffer

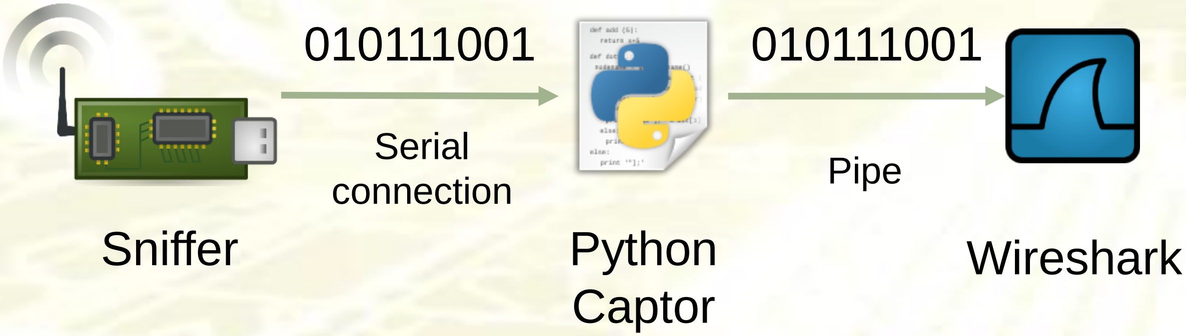
- Download sniffer program inside the example folder:
  - git clone https://github.com/lab-anaws/lab2-2017.git
- Load the sniffer into one sensor
  - make TARGET=sky MOTE=1 sniffer.upload

Sniffer

- Run the captor program:

python sensniff.py --non-interactive -d /dev/ttyUSB0

USB port of the mote acting as sniffer



Run Wireshark

- Run Wireshark
- Configure the program to collect packets from the mote:
  - Go to Capture -> options -> Manage Interfaces -> New (under Pipes) -> type /tmp/sensniff and save
  - The pipe will then appear as an interface. Start a capture on it

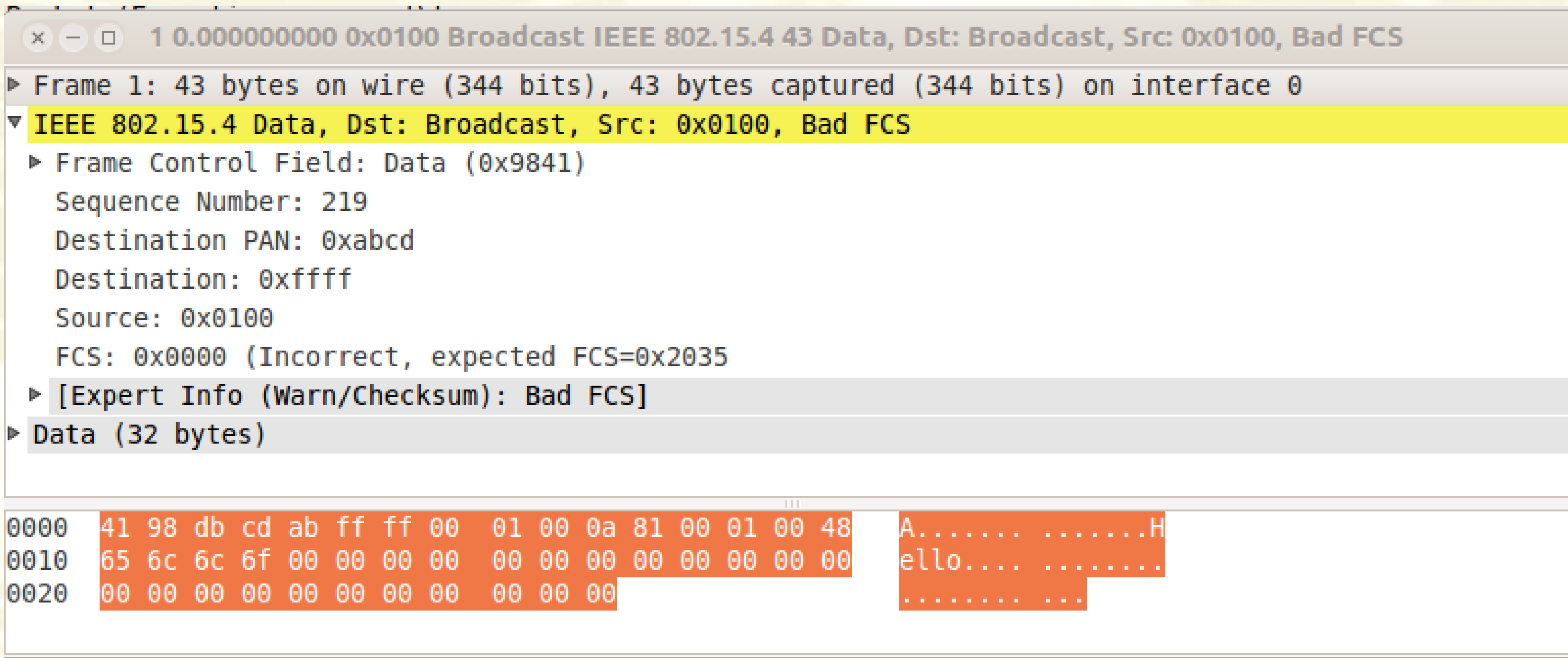
Generate some traffic

- Load on another mote a program to generate some traffic, e.g. “broadcast-example.c” in “examples/rime”

Set the same channel on all the motes!!

Captured data

- Captured data is shown in wireshark



Bad FCS Error

- Frame payload is not dissected (wireshark is supposed to analyze packets’ payload and show their content)
- An error, “Bad FCS”, is shown
- The frame check sequence (FCS) is a field included in IEEE802.15.4 frames to verify the integrity of the MAC frame
- *That field is processed in hardware*

Bad FCS Error

- Go to Edit -> Preferences
- Select Protocols -> IEEE 802.15.4
- Uncheck “Dissect only good FCS”