



*the sounds of smart environment*



## ***WP1 Acoustic Test bed qualification***

Qualify and Benchmark Test-beds for Acoustics in Deployment of Targeted Applications

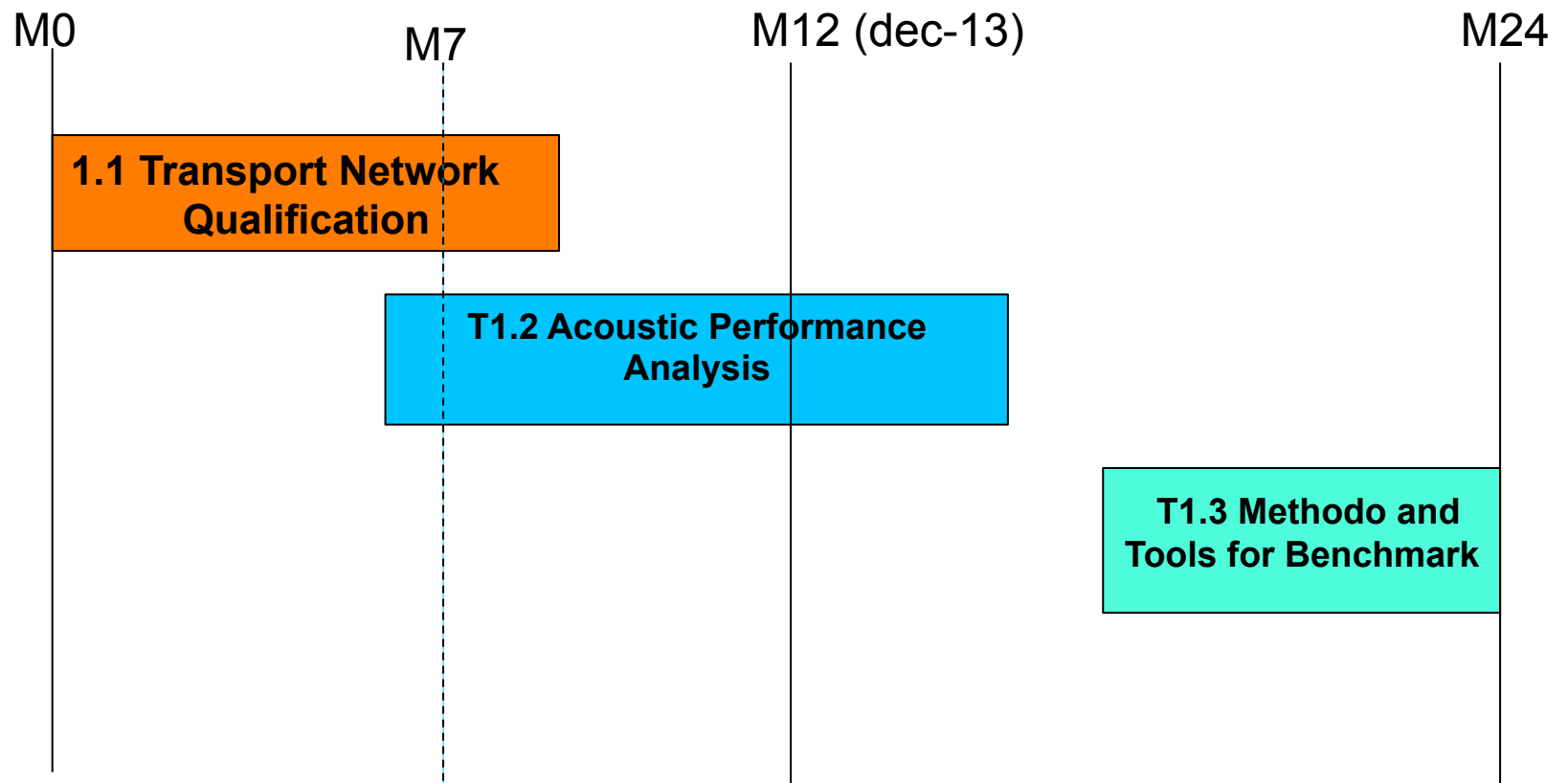
C. Pham (EGM & LIUPPA/University of Pau) and P. Cousin (EGM)



**MANDAT  
INTERNATIONAL**



# WP1 tasks





*the sounds of smart environment*



## ***T1.2 Acoustic Performance Analysis***

Methodology, lab tests and in-situ tests

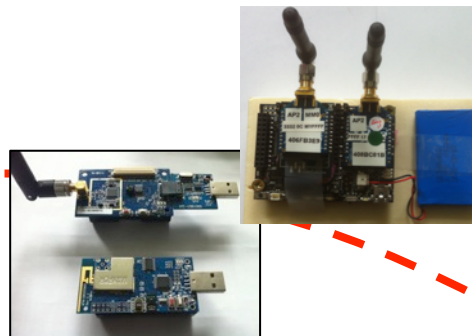


**MANDAT  
INTERNATIONAL**

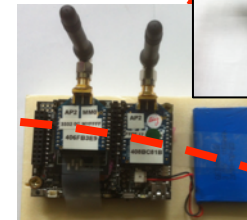
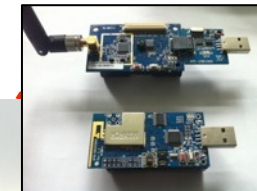


- Objectives- While network condition well established under previous task and links established to allow audio data stream, this part will investigate the minimum requirements and quality necessary for the exploitation of audio data as well as repeatability of the experiments. This will be done by specific audio measurement to qualify the environment and this will performed in close coordination with WP2 and WP3.
- Work plan- The general work plan for achieving the objectives of this task is:
  - Prepare for audio tests: Defined the condition where audio data will be collected and adapt if necessary the sensor for acoustic measurements;
  - Do audio-on-IP tests: perform test campaigns to collect audio-on-ip measurements on several different settings on the Santander and Geneva test sites. Measurement data can be on throughput, latency. Jitter, Packet loss rate, Packet loss patterns;
  - Provide overall data and analysis: provide several data and analysis to be used for benchmarking and could also lead to MOS (Mean Opinion Score).

# The sounds of smart environments



RELAY



PLAY/STORE RECEIVED  
AUDIO DATA



# Performance indicators

- Network performance indicators
  - 1-hop latency, relay latency, end-to-end latency
  - Packet jitter, packet loss rate
- Audio quality indicators
  - Can use Mean Opinion Score (MOS) to have a quantitative value to rank audio quality



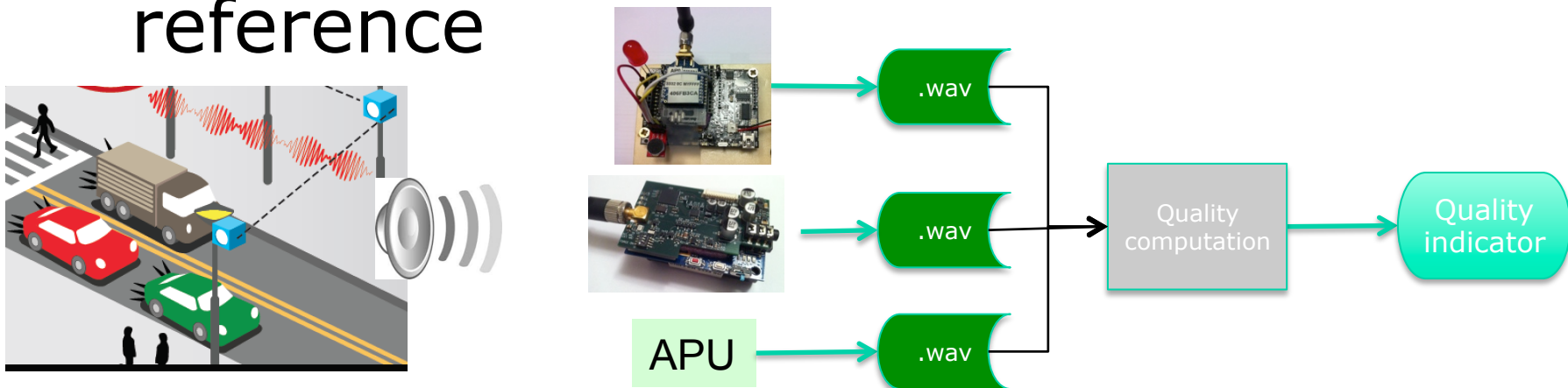
# Audio test-bed description



- Refer to document
  - [Audio test-bed description](#)

# Define audio reference quality

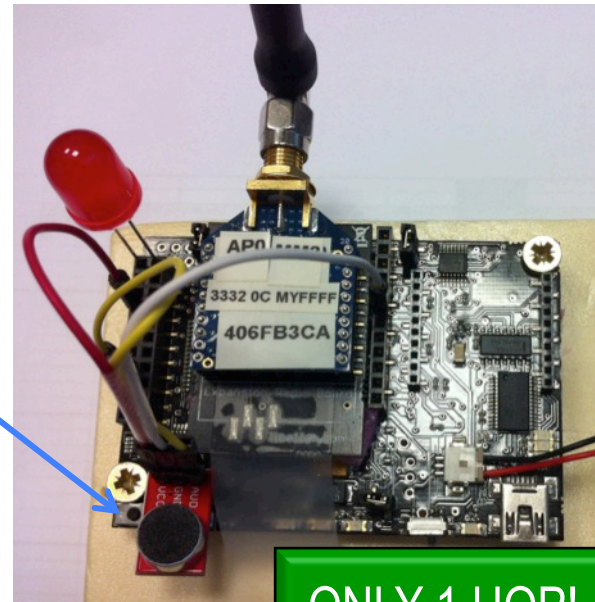
- Audio Processing Unit (APU) have better audio capture capabilities than IoT nodes
- Need to compare the quality of IoT's audio and APU's audio taken as a reference





# 1-hop: XBee in raw mode

- Electret mic with amplifier
- XBee in AP0 mode (transparent mode)
- 8-bit 4Khz sampling gives 32000bps
- 8Khz sampling gives 64000bps, requires custom API



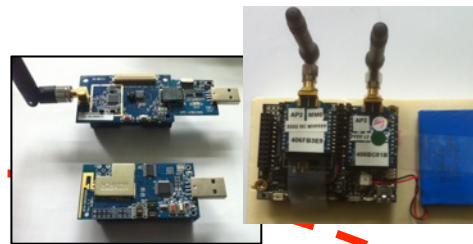
**ONLY 1 HOP!**



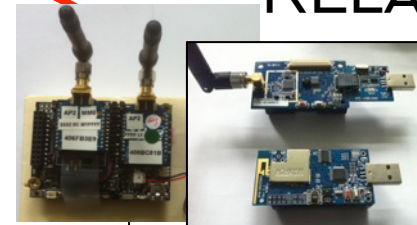
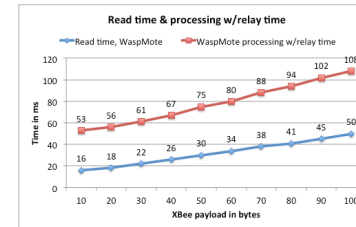
# Multi-hop audio constraints



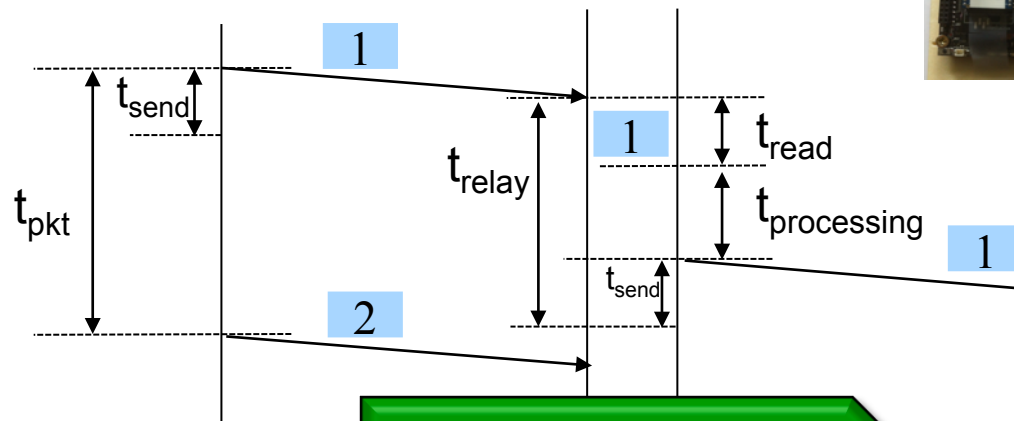
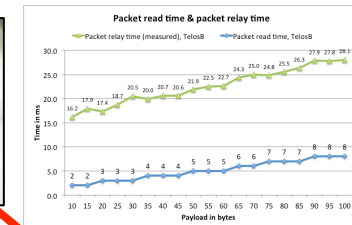
SOURCE



RELAY



RELAY



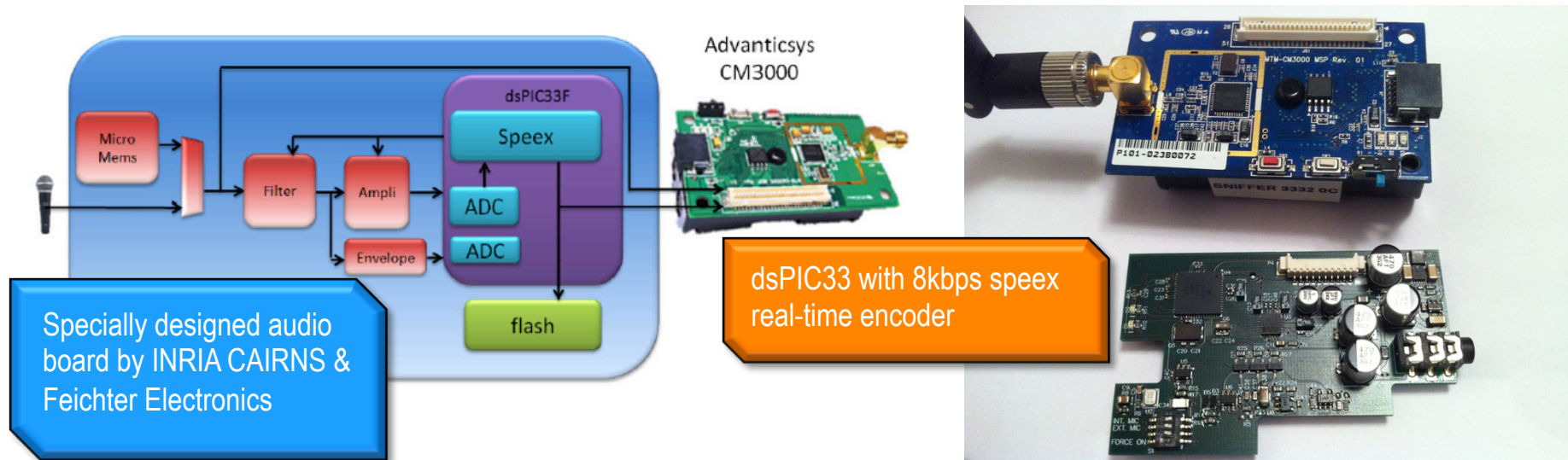
Intermediate nodes can have limited performances

PLAY RECEIVED FILE



# Multi-hop audio solution

- Use dedicated audio board for sampling/storing/encoding at 8kbps

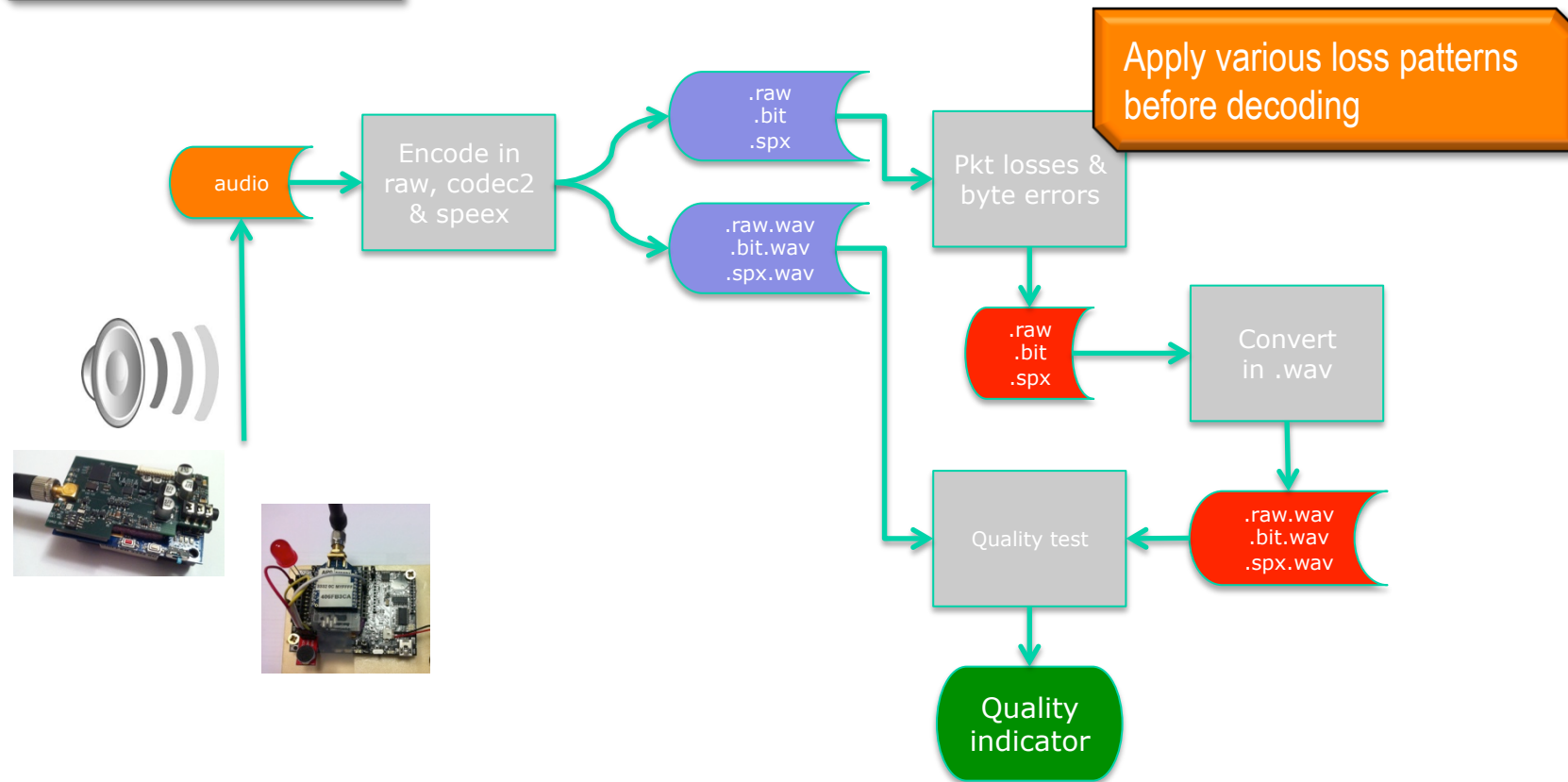


- Allows for multi-hop, encoded audio streaming scenarios

# Benchmark methodology

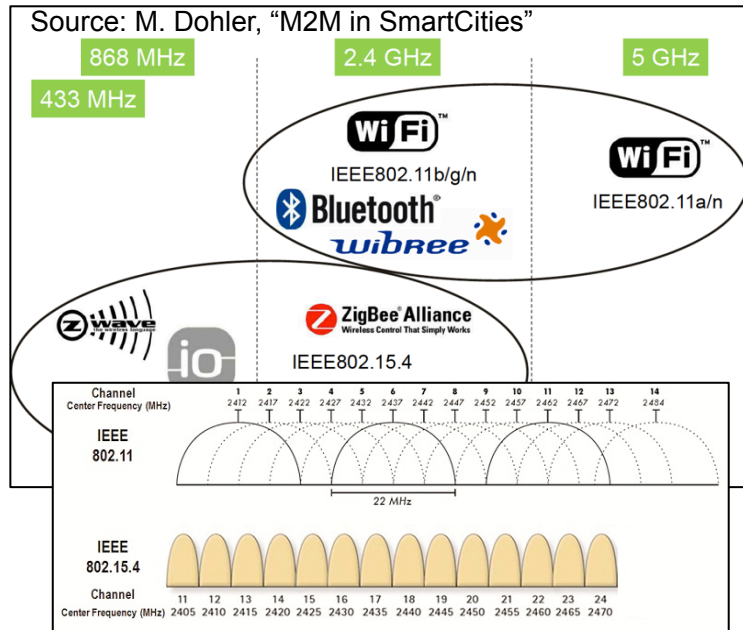
1. Determine sensitivity of codec against packet losses, with various packet size, lab tests
  - audio benchmarking, apply controlled packet error rates
  - MOS-LQO computation
2. Determine channel condition in selected areas, in-situ tests
  - Synthetic workload to determine packet loss rates
3. Determine latencies and jitter in multi-hop scenario, lab tests & in-situ tests
  - Controlled transmission of packetized/encoded audio
  - Measure latencies and jitter at intermediate nodes

## LAB TESTS



# Channel condition in selected areas

## IN-SITU TESTS



- Use representative locations in Santander & Geneva buildings
- Deploy IoT nodes traffic generators & sniffers
- Vary 802.15.4 channel and determine packet loss rates at various workload

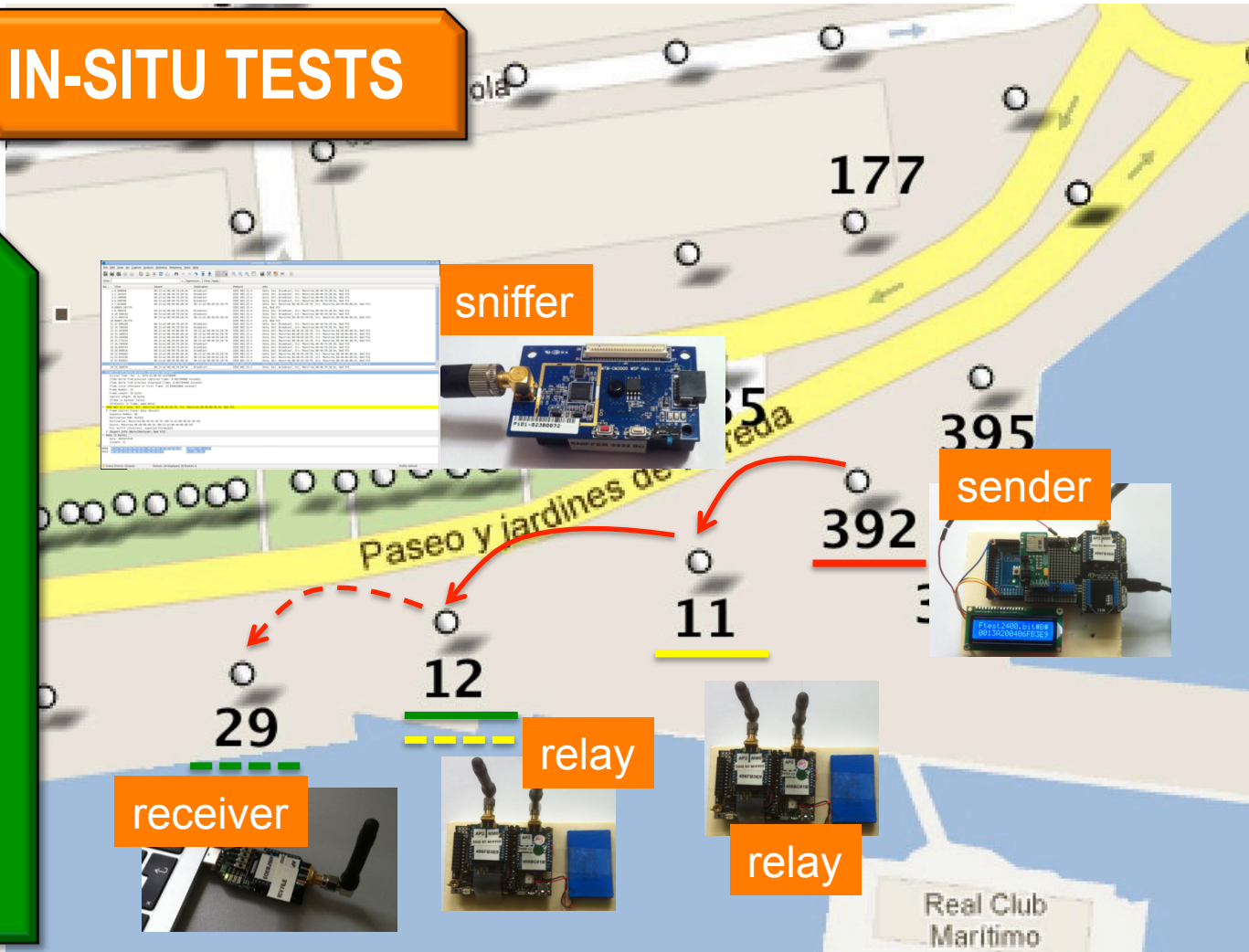
# Latency and jitter in multi-hop

## LAB TESTS

## IN-SITU TESTS

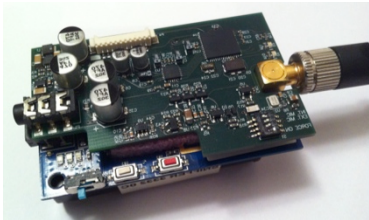
Sniffer node will capture all frames in order to measure inter-node latencies

Jitter will be measured at intermediate node as inter-packet time in known at sender side



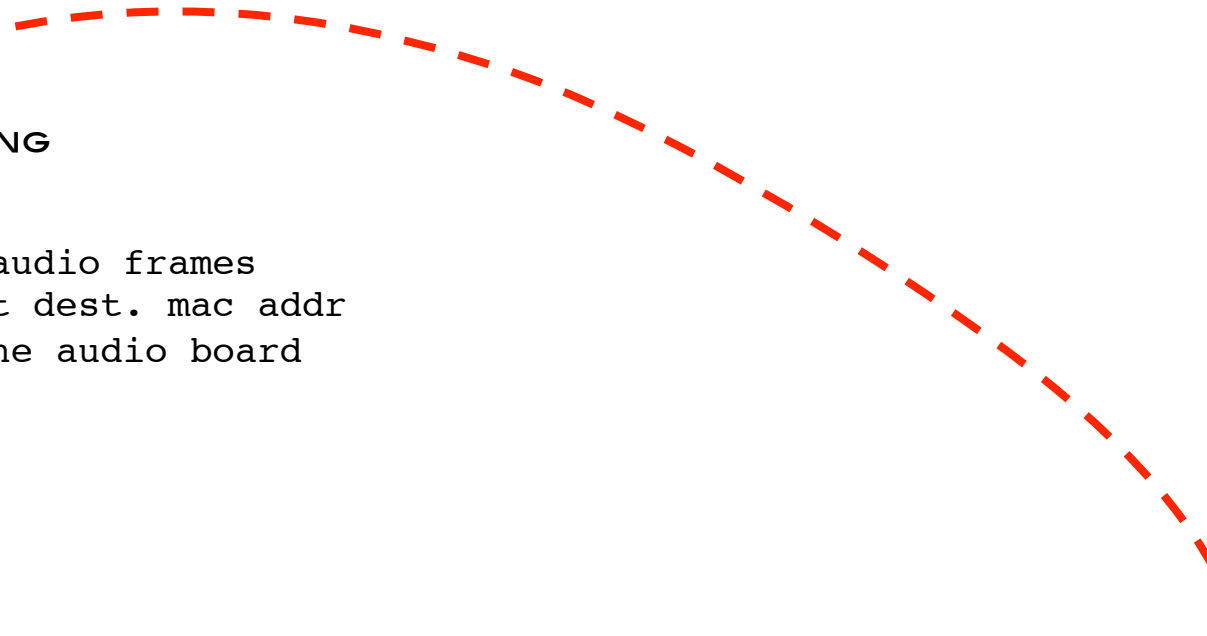
# 1-hop test-bed w/audio board

0x0090

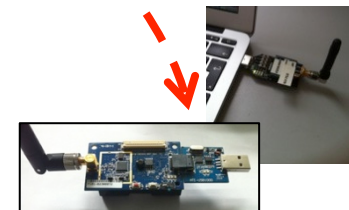


**SPEEX AUDIO ENCODING  
8KBPS**

**A**1/2/3/4 aggregate audio frames  
**D**0100 set the 16-bit dest. mac addr  
**C**0/1 power off/on the audio board



0x0100



```
python 115200SerialToStdout.py | ./speex_sampledec_wframing essai.raw |  
play --buffer 100 -t raw -r 8000 -s -2 -
```

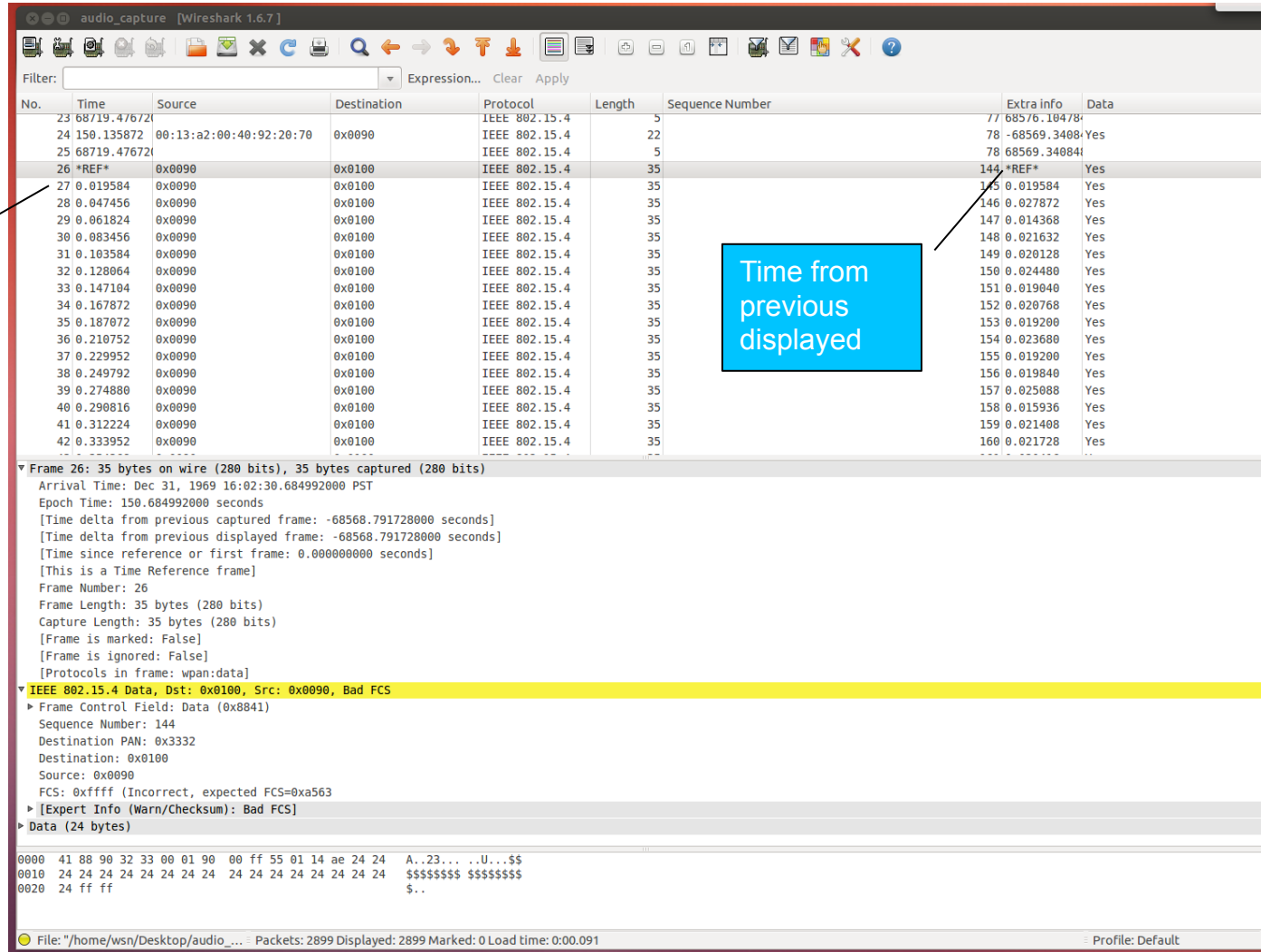
**DECODE & PLAY  
RECEIVED AUDIO**



# Example: latency 1-hop

Time from reference time

Time from previous displayed



No.	Time	Source	Destination	Protocol	Length	Sequence Number	Extra info	Data
23	68719.47672			IEEE 802.15.4	5		77 68576.10478	
24	150.135872	00:13:a2:00:40:92:20:70	0x0090	IEEE 802.15.4	22		78 -68569.3408	Yes
25	68719.47672			IEEE 802.15.4	5		78 68569.34084	
26	*REF*	0x0090	0x0100	IEEE 802.15.4	35		144 *REF*	Yes
27	0.019584	0x0090	0x0100	IEEE 802.15.4	35		145 0.019584	Yes
28	0.047456	0x0090	0x0100	IEEE 802.15.4	35		146 0.027872	Yes
29	0.061824	0x0090	0x0100	IEEE 802.15.4	35		147 0.014368	Yes
30	0.083456	0x0090	0x0100	IEEE 802.15.4	35		148 0.021632	Yes
31	0.103584	0x0090	0x0100	IEEE 802.15.4	35		149 0.020128	Yes
32	0.128064	0x0090	0x0100	IEEE 802.15.4	35		150 0.024480	Yes
33	0.147104	0x0090	0x0100	IEEE 802.15.4	35		151 0.019040	Yes
34	0.167872	0x0090	0x0100	IEEE 802.15.4	35		152 0.020768	Yes
35	0.187072	0x0090	0x0100	IEEE 802.15.4	35		153 0.019200	Yes
36	0.210752	0x0090	0x0100	IEEE 802.15.4	35		154 0.023680	Yes
37	0.229952	0x0090	0x0100	IEEE 802.15.4	35		155 0.019200	Yes
38	0.249792	0x0090	0x0100	IEEE 802.15.4	35		156 0.019840	Yes
39	0.274880	0x0090	0x0100	IEEE 802.15.4	35		157 0.025088	Yes
40	0.290816	0x0090	0x0100	IEEE 802.15.4	35		158 0.015936	Yes
41	0.312224	0x0090	0x0100	IEEE 802.15.4	35		159 0.021408	Yes
42	0.333952	0x0090	0x0100	IEEE 802.15.4	35		160 0.021728	Yes

▼ Frame 26: 35 bytes on wire (280 bits), 35 bytes captured (280 bits)  
 Arrival Time: Dec 31, 1969 16:02:30.684992000 PST  
 Epoch Time: 150.684992000 seconds  
 [Time delta from previous captured frame: -68568.791728000 seconds]  
 [Time delta from previous displayed frame: -68568.791728000 seconds]  
 [Time since reference or first frame: 0.000000000 seconds]  
 [This is a Time Reference frame]  
 Frame Number: 26  
 Frame Length: 35 bytes (280 bits)  
 Capture Length: 35 bytes (280 bits)  
 [Frame is marked: False]  
 [Frame is ignored: False]  
 [Protocols in frame: wlan:data]

▼ IEEE 802.15.4 Data, Dst: 0x0100, Src: 0x0090, Bad FCS  
 ▶ Frame Control Field: Data (0x8841)  
 Sequence Number: 144  
 Destination PAN: 0x3332  
 Destination: 0x0100  
 Source: 0x0090  
 FCS: 0xffff (Incorrect, expected FCS=0xa563)  
 ▶ [Expert Info (Warn/Checksum): Bad FCS]  
 ▶ Data (24 bytes)

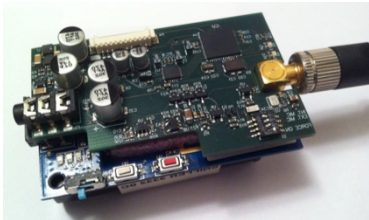
```

0000 41 88 90 32 33 00 01 90 00 ff 55 01 14 ae 24 24  A..23...U...$
0010 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24  $$$$$$$$
0020 24 ff ff                                     $..
  
```

File: "/home/wsn/Desktop/audio\_... : Packets: 2899 Displayed: 2899 Marked: 0 Load time: 0:00.091 Profile: Default

# 2-hop test-bed w/audio board

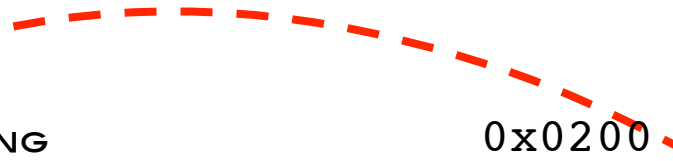
0x0090



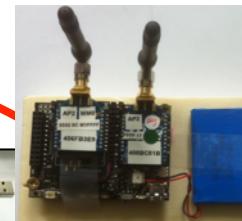
**SPEEX AUDIO ENCODING  
8KBPS**

**A**1/2/3/4/6 aggregate audio frames  
**D**0200 set the 16-bit dest. mac addr  
**C**0/1 power off/on the audio board

**R**0/1 enable/disable relay mode  
**D**0100 set the 16-bit dest. mac addr

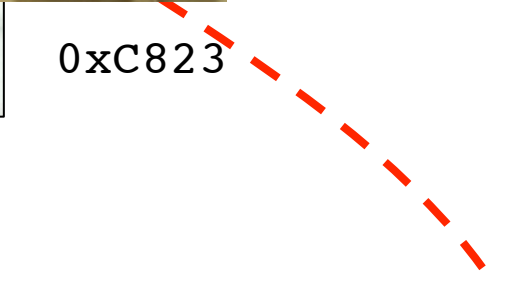


0x0200

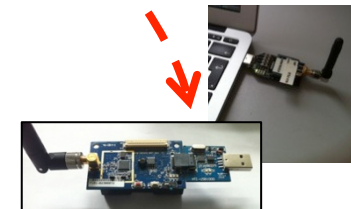


**RELAY**

0xC823



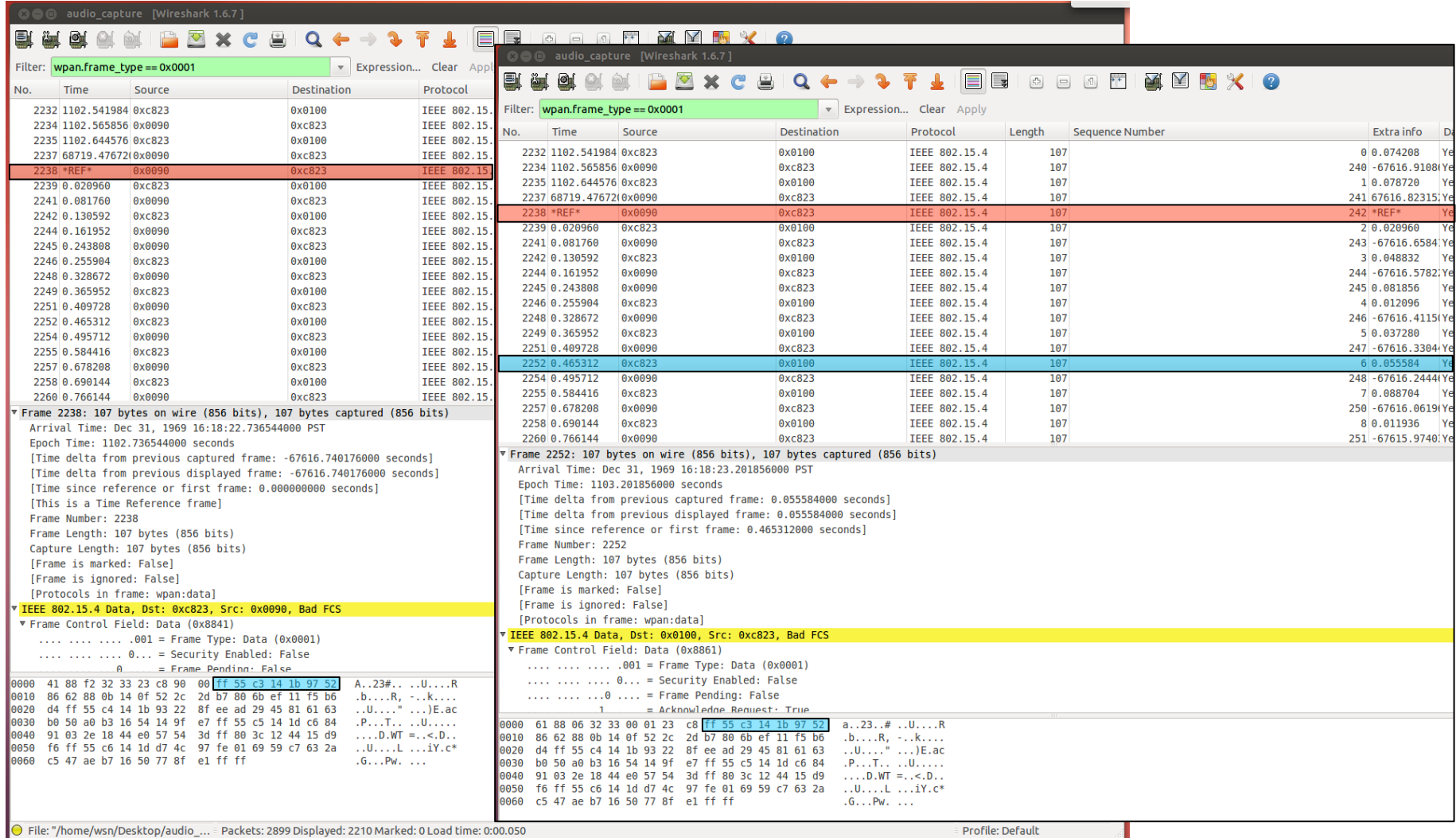
0x0100



**DECODE & PLAY  
RECEIVED AUDIO**

```
python 115200SerialToStdout.py | ./speex_sampledec_wframing essai.raw |
play --buffer 100 -t raw -r 8000 -s -2 -
```

# Example: latency 2-hop

The image displays two screenshots of the Wireshark network protocol analyzer. Both screenshots show a packet capture with the filter `wpan.frame_type == 0x0001`.

**Left Screenshot:** Shows a list of captured packets. Packet 2238 is highlighted in red, indicating a 'Bad FCS' (Frame Check Sequence) error. The packet details pane shows the IEEE 802.15.4 Data section with the following information:

- IEEE 802.15.4 Data, Dst: 0xc823, Src: 0x0090, Bad FCS
- Frame Control Field: Data (0x8841)
- Frame Type: Data (0x0001)
- Security Enabled: False
- Frame Pending: False

**Right Screenshot:** Shows a detailed view of a frame (No. 2252) with IEEE 802.15.4 Data, Dst: 0x0100, Src: 0xc823, Bad FCS. The packet details pane shows the IEEE 802.15.4 Data section with the following information:

- IEEE 802.15.4 Data, Dst: 0x0100, Src: 0xc823, Bad FCS
- Frame Control Field: Data (0x8861)
- Frame Type: Data (0x0001)
- Security Enabled: False
- Frame Pending: False
- Acknowledge Request: True

Both screenshots also show the raw packet bytes at the bottom of the packet details pane.

# Audio quality: PESQ & MOS (1)

- ITU-T P.862 Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs.
- Download software at :  
[https://www.itu.int/rec/dologin\\_pub.asp?lang=e&id=T-REC-P.862-200511-I!Amd2!SOFT-ZST-E&type=items](https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-P.862-200511-I!Amd2!SOFT-ZST-E&type=items)

# Audio quality: PESQ & MOS (2)

- We can use ITU-T PESQ tool to determine the MOS value for loss-free encoded audio (codec2, speex). MOS-LQO values greater than 2.6 are considered quite acceptable
- 5=Excellent, 4=Good, 3=Fair, 2=Poor, 1=Bad

REFERENCE	DEGRADED	PESQMOS	MOSLQO	SAMPLE_FREQ
test.wav	test.wav	4.500	4.549	8000
test.wav	test2150.spx.wav	2.757	2.472	8000
test.wav	test5950.spx.wav	3.428	3.454	8000
test.wav	test8000.spx.wav	3.652	3.757	8000
test.wav	test11000.spx.wav	3.941	4.093	8000
test.wav	test13000.spx.wav	3.941	4.093	8000
test.wav	test15000.spx.wav	4.085	4.235	8000
test.wav	test1600.bit.raw.wav	2.648	2.323	8000
test.wav	test1400.bit.raw.wav	2.625	2.293	8000
test.wav	test2400.bit.raw.wav	2.768	2.487	8000
test.wav	test3200.bit.raw.wav	2.801	2.533	8000



# PESQ & MOS of iThing'13 results

REFERENCE	DEGRADED	PESQMOS	MOSLQO	SAMPLE_FREQ
test2400.bit.raw.wav	test2400-44-105-6L-F77.bit.raw.wav	2.752	2.465	8000
test2400.bit.raw.wav	test2400-44-110-0L.bit.raw.wav	4.500	4.549	8000
test2400.bit.raw.wav	test2400-54-110-5L-F77.bit.raw.wav	2.725	2.427	8000
test2400.bit.raw.wav	test2400-54-120-2L-F77.bit.raw.wav	3.239	3.178	8000
test2400.bit.raw.wav	test2400-64-120-5L-F77.bit.raw.wav	2.737	2.444	8000
test2400.bit.raw.wav	test2400-64-125-2L-F77.bit.raw.wav	3.689	3.804	8000
test2400.bit.raw.wav	test2400.bit.raw.wav	4.500	4.549	8000
test.wav	test2400-44-105-6L-F77.bit.raw.wav	2.600	2.260	8000
test.wav	test2400-44-110-0L.bit.raw.wav	2.768	2.487	8000
test.wav	test2400-54-110-5L-F77.bit.raw.wav	2.310	1.919	8000
test.wav	test2400-54-120-2L-F77.bit.raw.wav	2.648	2.323	8000
test.wav	test2400-64-120-5L-F77.bit.raw.wav	2.307	1.916	8000
test.wav	test2400-64-125-2L-F77.bit.raw.wav	2.679	2.365	8000
test.wav	test2400.bit.raw.wav	2.768	2.487	8000
test.wav	test.wav	4.500	4.549	8000