SIGFOX ultra-narrowband network optimization

Tomaž Šolc, Timotej Gale, Carolina Fortuna
tomaz.solc@ijs.si

Department of Communication Systems
Jožef Stefan Institute

Introduction

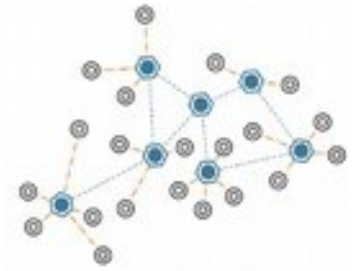
eWINE project

- Elastic Wireless Networking Experimentation
  - How to make wireless networks that scale on demand? (like hosting something on the Amazon cloud service)
  - Improvements on network, MAC and physical layers
- 2 year H2020 project (started Jan 2016)

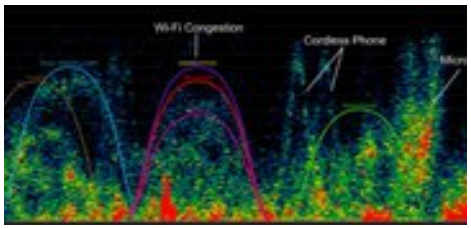
Heterogeneous technologies

Complex configuration interactions

Large scale networks



Network dynamics and external influences



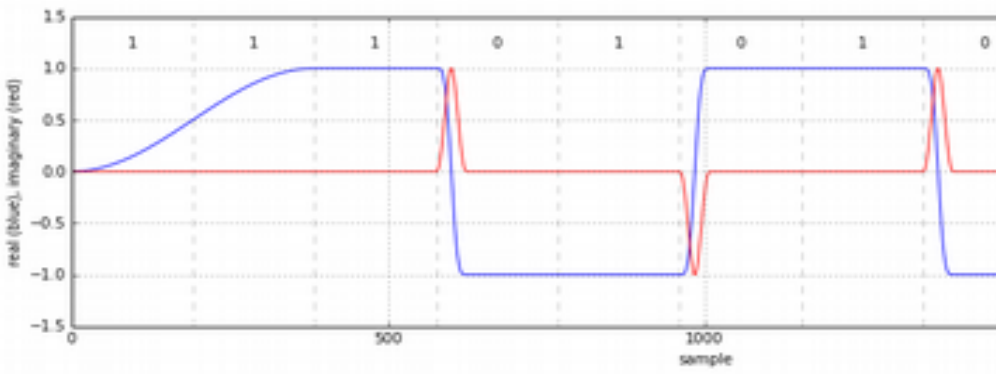
imec · Trinity College Dublin · is-wireless · MARTEL innovate · Jožef Stefan Institute · THALES · sigfox Make Things Come Alive · berlin · Technische Universität Dresden · Spacetime Networks

LP-WANs

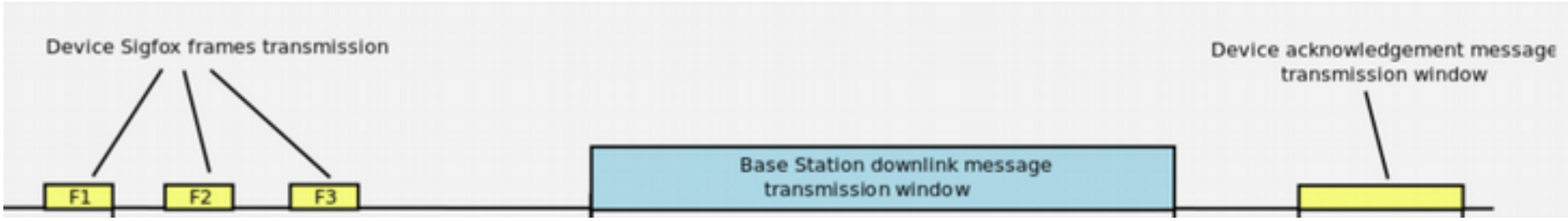
- Low Power Wide Area Networks
  - provide connectivity to small, battery powered devices (e.g. smart meters, sensors, Internet of Things, etc.)
  - >10 km cell coverage, >1 year on battery life, ~100 byte payloads, mostly uplink, high latency
  - **why not existing mobile networks?** 2G is being retired, 3G-4G focus on broadband,
- Many competing technologies and networks
  - LoRa/LoRaWAN, SIGFOX, Weightless, ...
  - LTE-M, NB-LTE release 13

SIGFOX network

- Ultra-narrowband physical layer
  - Using unlicensed bands (868 MHz in Europe)
  - 100 bits/s, DBPSK, 1500 microchannels



- Opportunistic media access
  - ALOHA protocol: on uplink device (pseudo-) randomly chooses time and frequency of transmission.
  - 3 frame retransmissions (on different channels).



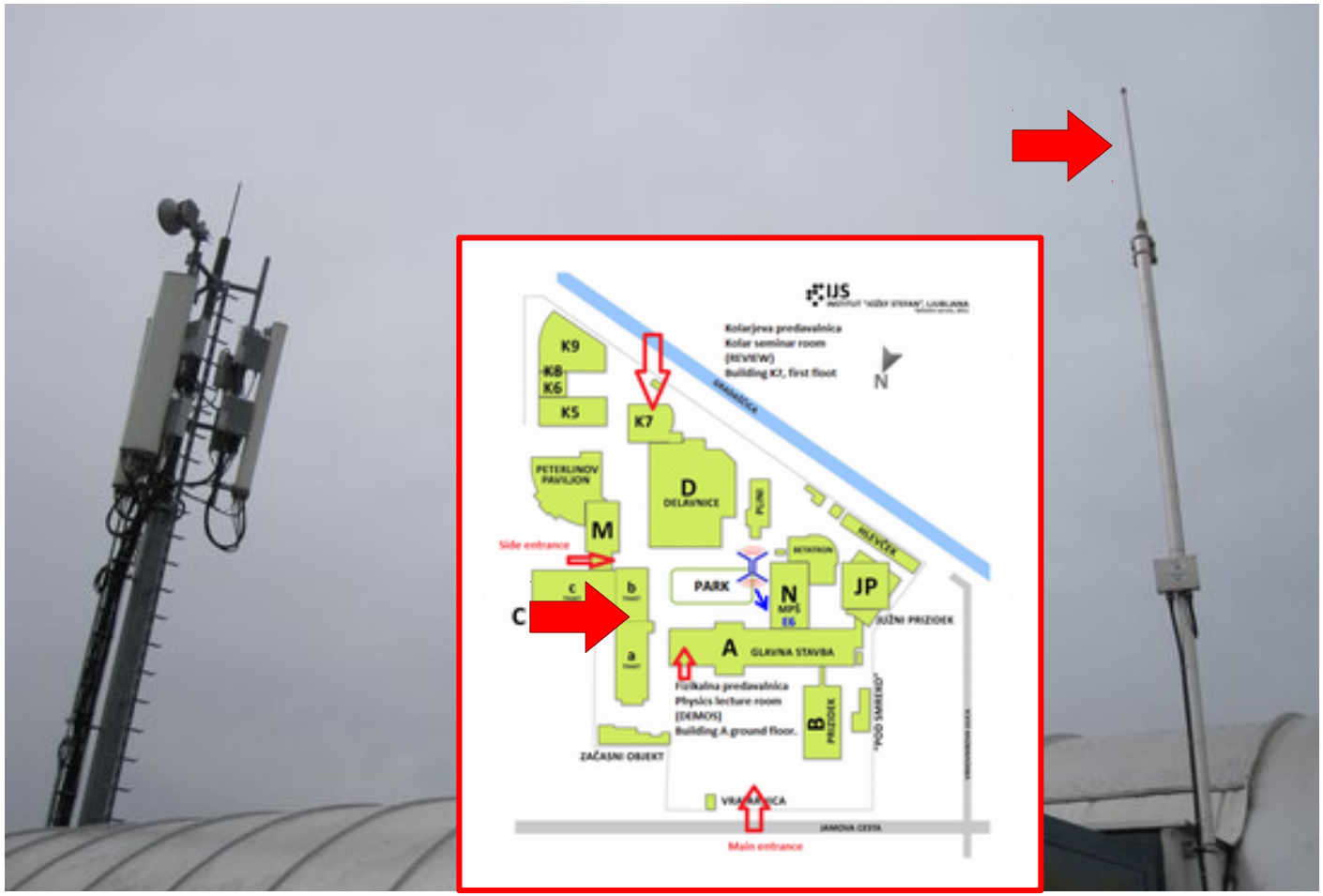
Motivation for our work

- Devices in LP-WANs face an interference problem in dense environments.
  - Inter-technology: Unlicensed band are shared
  - Intra-technology: ALOHA may not be optimal
- Can we design a testbed architecture for **rapid experimentation** with UNB technology?
- Improving the network: **spectrum** and **battery life**
  - Can we get same QoS with fewer frame retransmissions?
  - Can we devise a smarter channel selection algorithm than random choice?

Setup

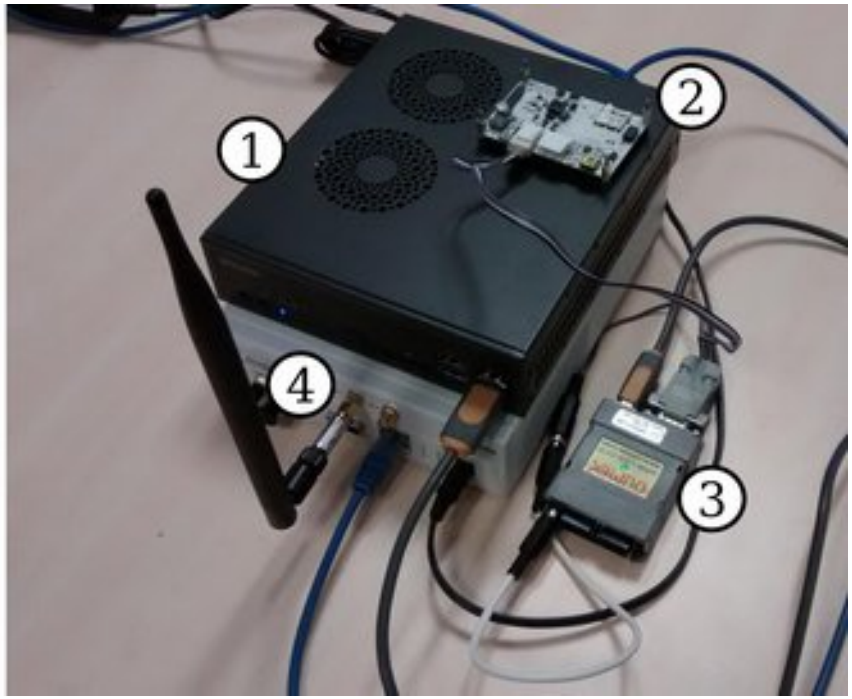
SIGFOX base station @ JSI

- On loan from SIGFOX, mounted in May 2016.
  - Did not require modifications for our work.



Experimental SIGFOX device

- We built our own device for rapid experimentation
  - Firmware in production modems is hard to modify.
  - We used a software-defined radio approach.
- To avoid re-implementing the MAC layer, we run part of the original modem firmware on an ARM CPU.



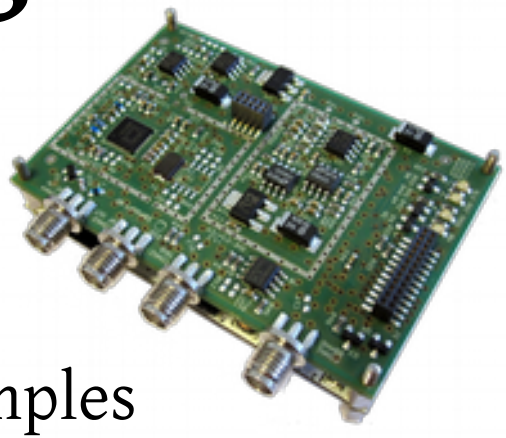
1) Compact PC running GNU/Linux OS and our SIGFOX PHY implementation using Python, numpy, GNU Radio framework.

2) ARM board running SIGFOX MAC layer.

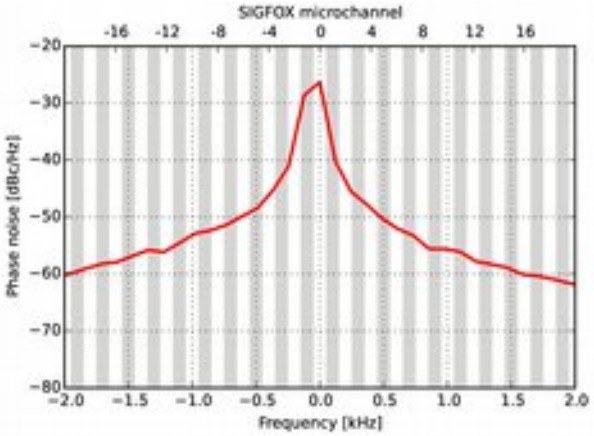
3) serial interface between ARM and PC

4) SDR front-end (Ettus USRP N210)

Spectrum sensing

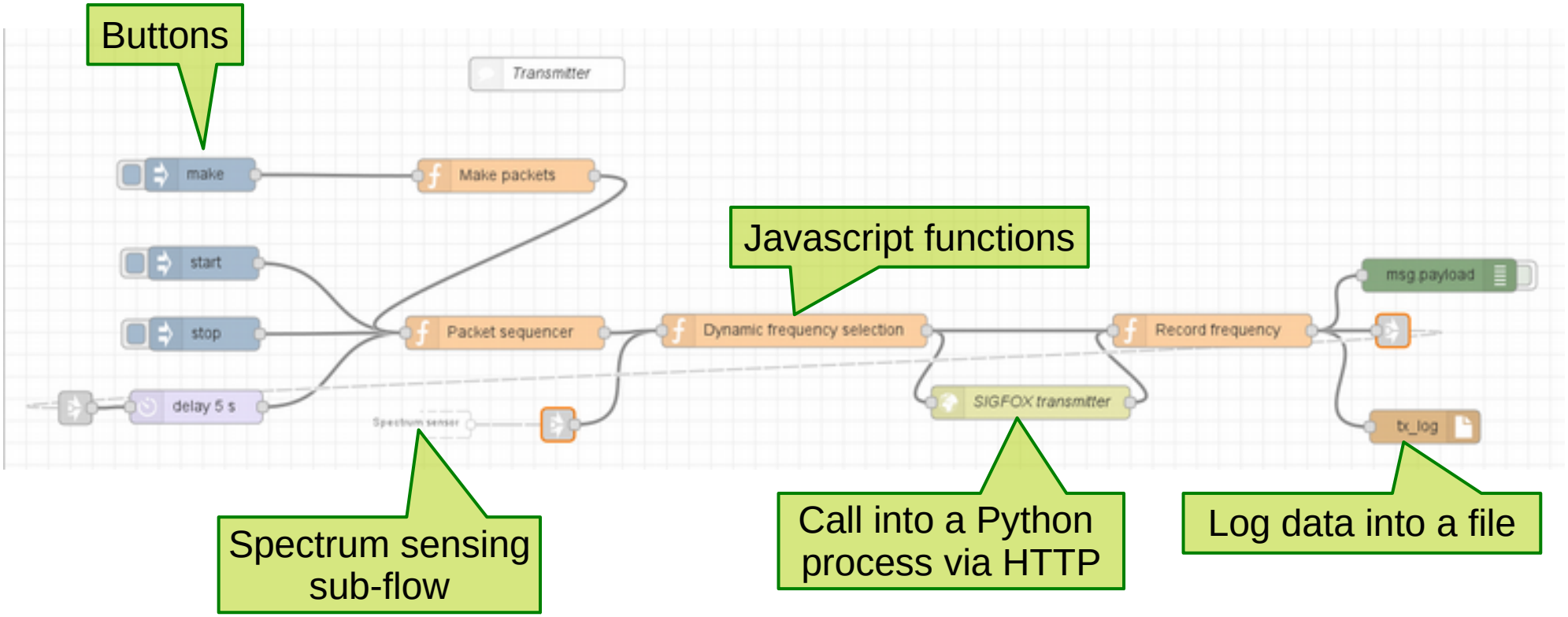


- VESNA SNE-ESHTER spectrum sensor
  - custom low cost receiver developed in 2015
  - UHF reception, up to 2 Msamples/s, 25k samples
  - one receiver mounted on roof nearby SIGFOX basestation
- Energy detection based on FFT of signal samples
  - 200 Hz resolution (one bin = 2 SIGFOX μchannels)
- Challenges
  - high phase noise (-58 dBc/Hz @ 1 kHz)
  - low sensitivity (868 MHz is on the edge of the antenna and receiver pass band)



Experiment controller

- Node-RED used as glue between components
  - Flow-based visual programming tool from IBM
  - Browser-based (experiments can be done remotely)

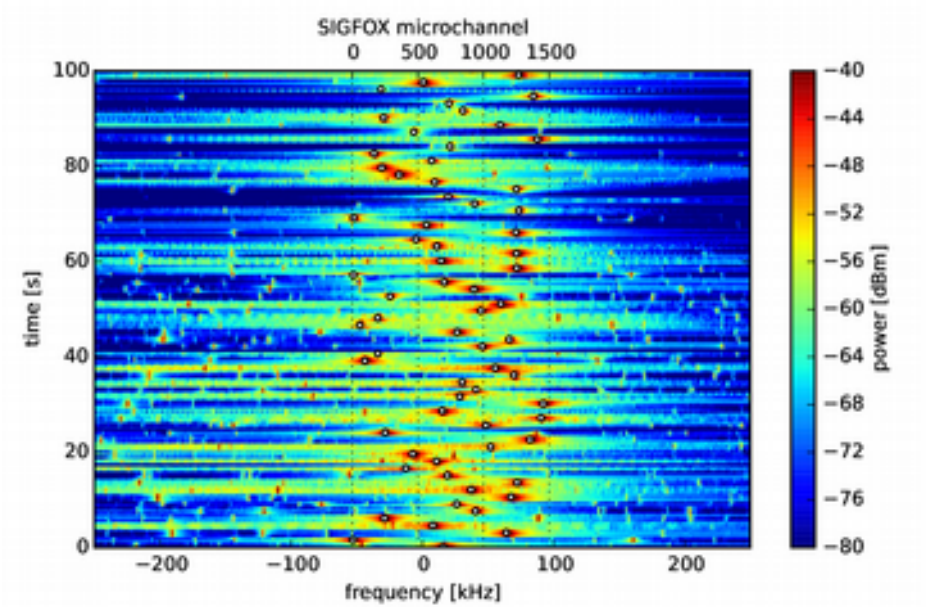


Experiments & results

Dataset collection

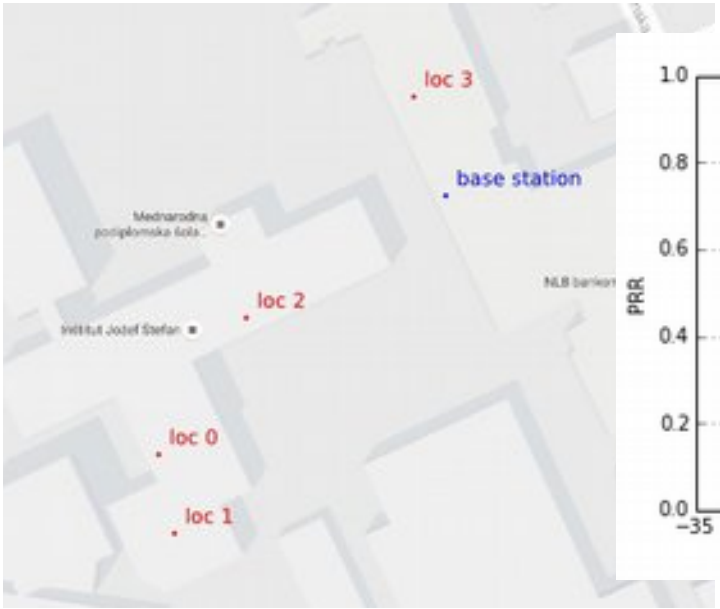
- Packet based data
  - RSSI, SNR as reported by the base station
  - Packet loss calculated from sequence num.
- Spectrum data
  - Power spectral density with 200 Hz resolution
  - Time synchronized with packet data

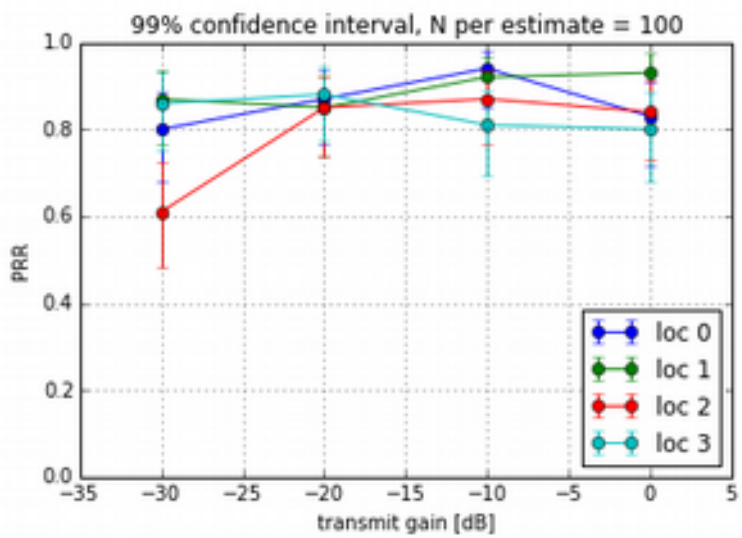

```
{
    "seqNumber": "675",
    "avgSnr": "10.06",
    "station": "0BF2",
    "snr": "11.99",
    "time": "1473675863",
    "device": "1CF14C",
    "rssi": "-129.00",
    "data": "0001"
},
```

Dataset collection

- Currently collected data contains ~30k packets
  - various device locations, transmit powers
  - frame repetition patterns, ch. selection algorithms, ...
- Some datasets already published on GitHub, want to eventually submit to CRAWDAD, etc.



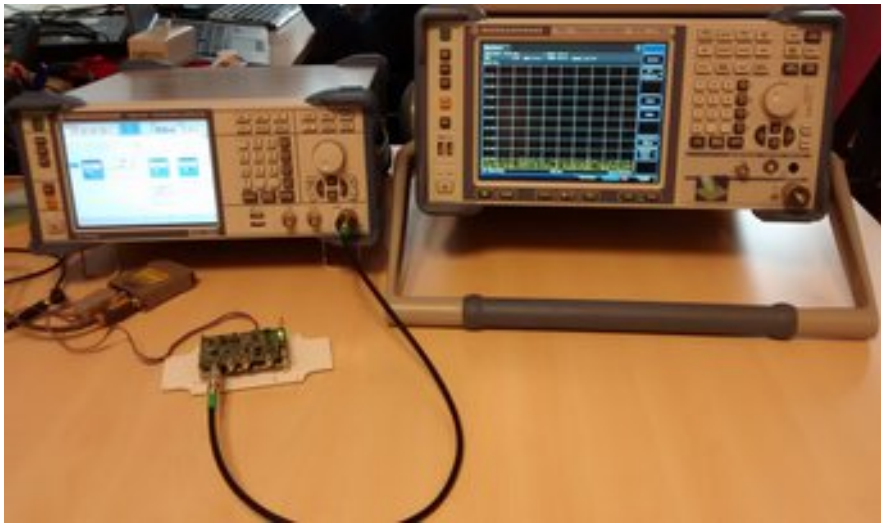


Uplink optimization

- Increase **uplink packet reception ratio** by intelligently choosing the transmit channel.
- How to find most vacant channels?
  - **physical layer**: channel occupancy table
  - **MAC layer**: link quality estimation and prediction
- Time frame for optimization?
  - **fast loop**: do channel assessment on device, transmit immediately when channel is vacant (e.g. LBT)
  - **slow loop**: do channel assessment on network, occasionally transmit statistics to devices.

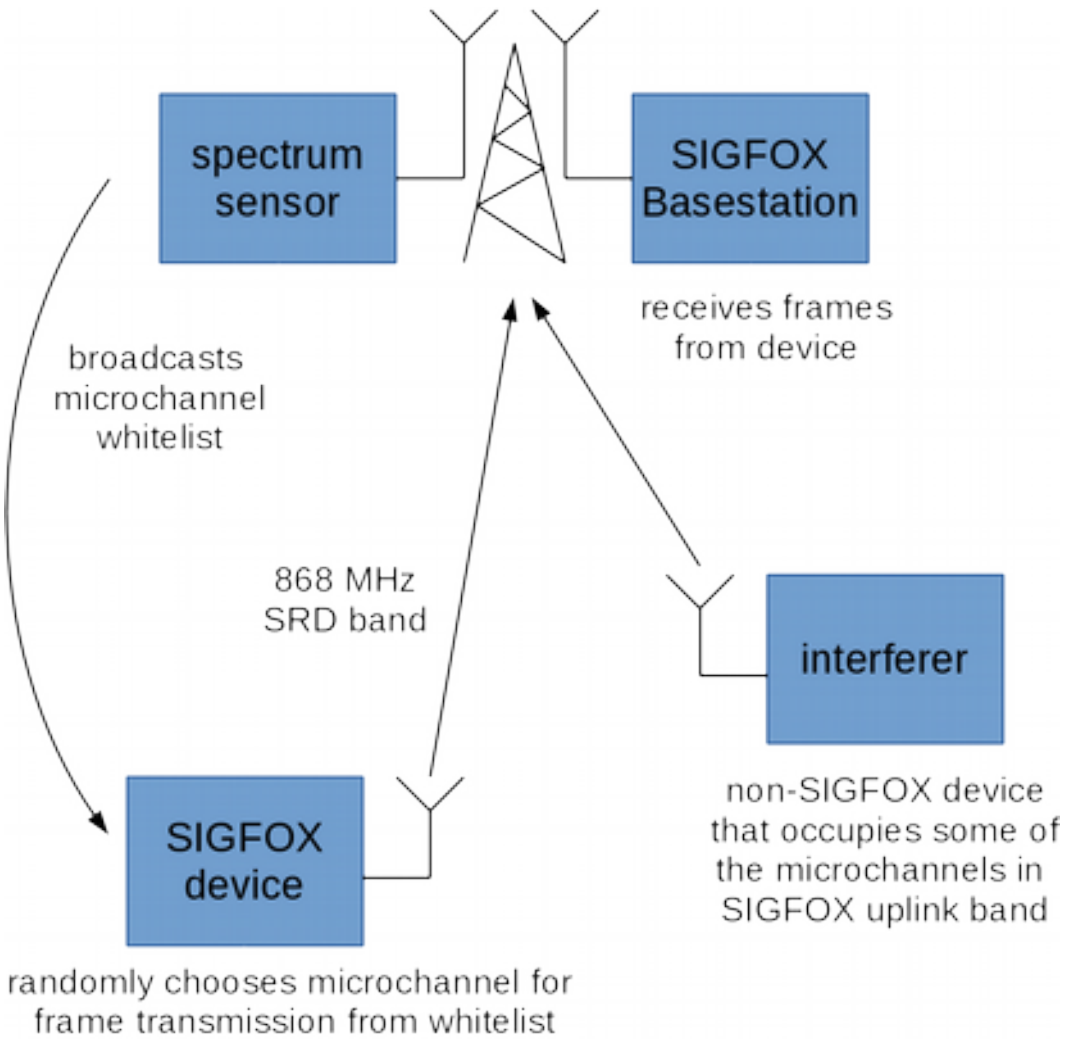
Selecting least occupied channel

- Mean PSD over time window got best results.
- Experimentally evaluated covariance detectors.
  - ~3 dB better sensitivity than ED @ Pfa = 5%, Pd = 99%
  - very long sensing time required for 100 Hz resolution
- Energy detection with ROHT algorithm.
  - Occupied/vacant decision based on dynamically adjusted thresholds.
  - Doesn't seem to work well with UNB signals.



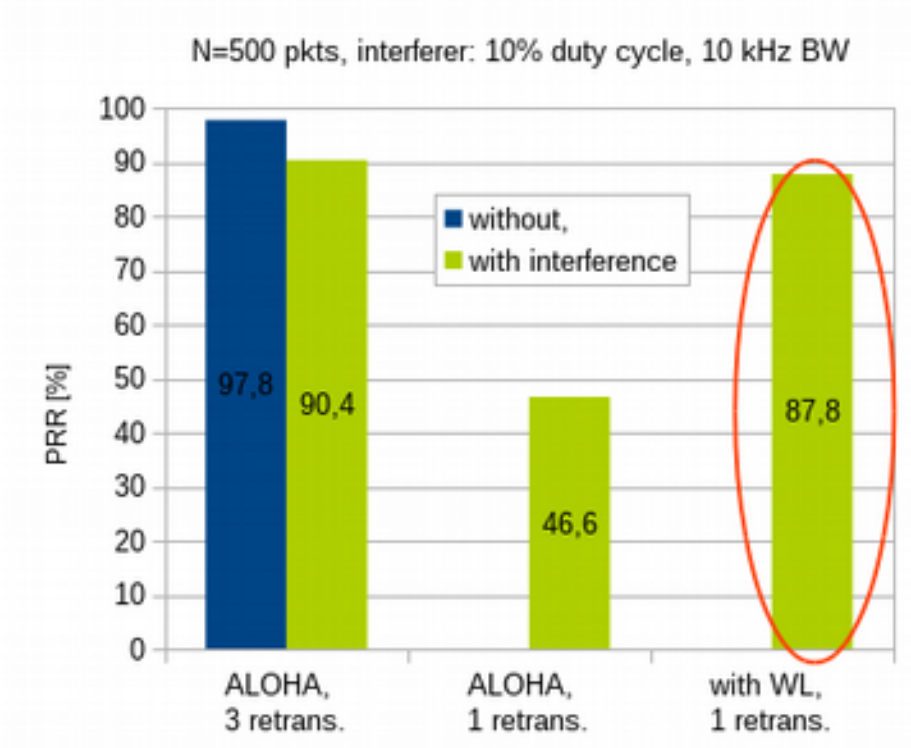
Proposed dynamic channel selection scheme

- Interference avoidance for **uplink transmissions**
- **Slow loop approach**
  - Devices don‘t sense (saves power vs. LBT)
  - Whitelist BS broadcast saves spectrum - O(1) vs. O(n) with retransmissions
- Based on energy detection
- **Goal**: Same PRR with <3x frame repetitions in dense, heterogeneous environments



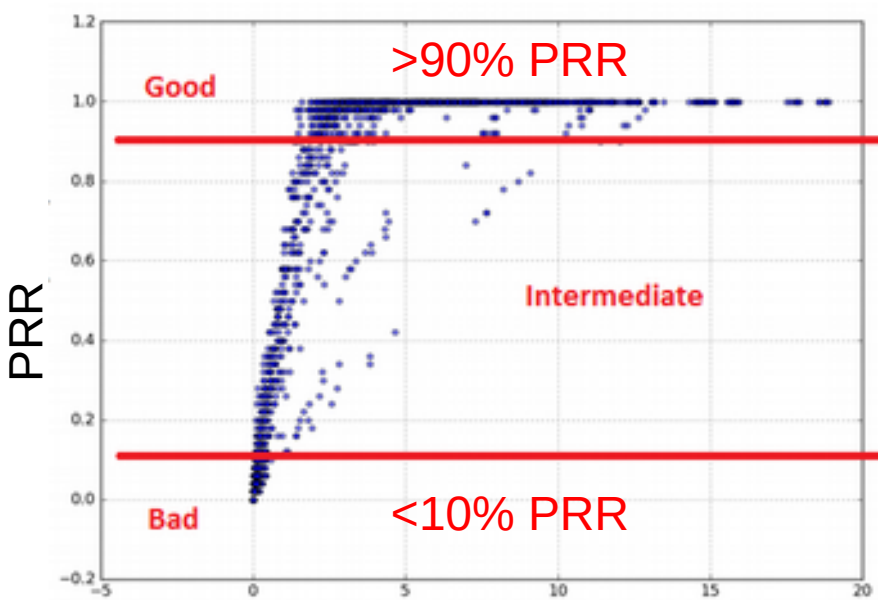
Experiment with interference

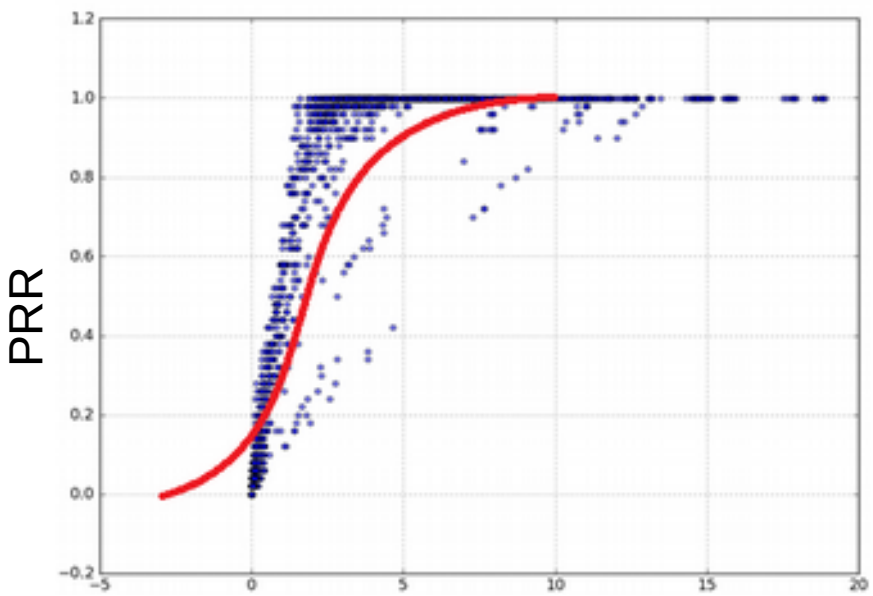
- Artificial generated interference in the uplink band
  - simulated packet-based OFDM transmission, 10% duty cycle, occupying 50% of available microchannels
- Compare packet reception rate: **ALOHA vs whitelisting**
  - whitelist 50 microchannels with lowest mean PSD
- **Only ~3 pp. drop in PRR**
  - 1 transmission and whitelisting vs.
  - 3 transmissions and existing SIGFOX ALOHA protocol



Link quality estimation

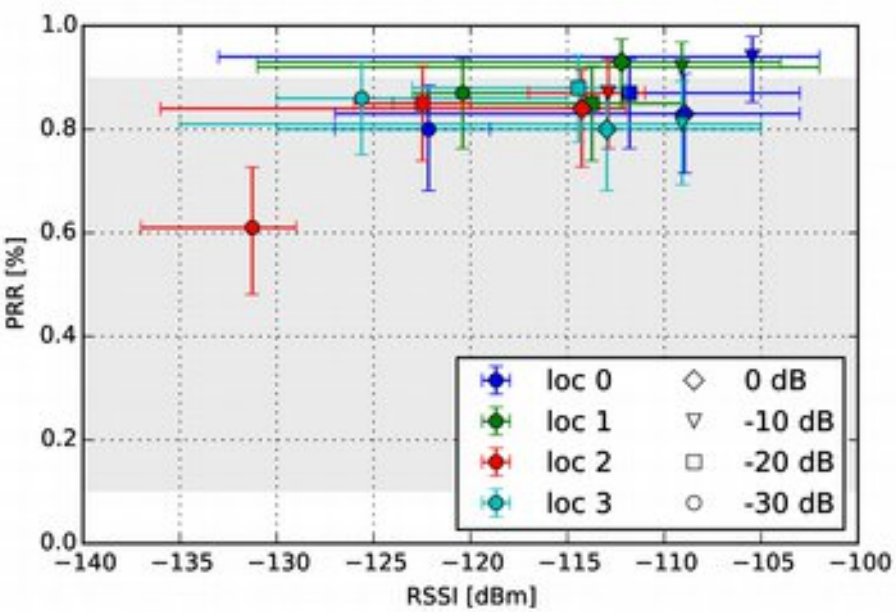
- Can we **predict** which the best link in the future?
  - Mean, variance of **RSSI, SNR, PRR** over time window.
- Classification or regression?
  - The usefulness and accuracy of LQE is a debated topic.





Link quality estimation

- Training data
  - 4 locations, 4 TX powers
  - 16 x 100 = 1600 packets
  - 37% good links,
    63% intermediate links



- Classification results
  - WEKA machine learning
  - J48 decision tree

TABLE I. LQE CLASSIFICATION FOR THE BEST PERFORMING FEATURE VECTORS.

Feature vector	Correct	Incorrect
rssi, avg(rssi), std(rssi), avg(snr), std(snr), avgSnr	78.82%	21.17 %
avg(rssi), std(rssi), std(snr), avgSnr	78.01 %	21.98%
avg(rssi), avg(snr), std(snr), avgSnr	77.42%	22.57 %

Link quality estimation

- Comparing with datasets from CRAWDAD
  - IEEE 802.11 from Rutgers University

		Resampled	Interpolated	Interpolated and undersampled
RSSI, RSSI avg, RSSI std	Correct class	**77.1956** %	**95.7686** %	**88.6853** %
	Incorrect class	**22.8044** %	**4.2314** %	**11.3147** %

  - IEEE 802.11 from Colorado Universitys
    - missing sequence numbers.
  - IEEE 802.15.4 from Michigan University
    - unable to generate labels for the classification tasks.

Future work

Problems with LTE interference

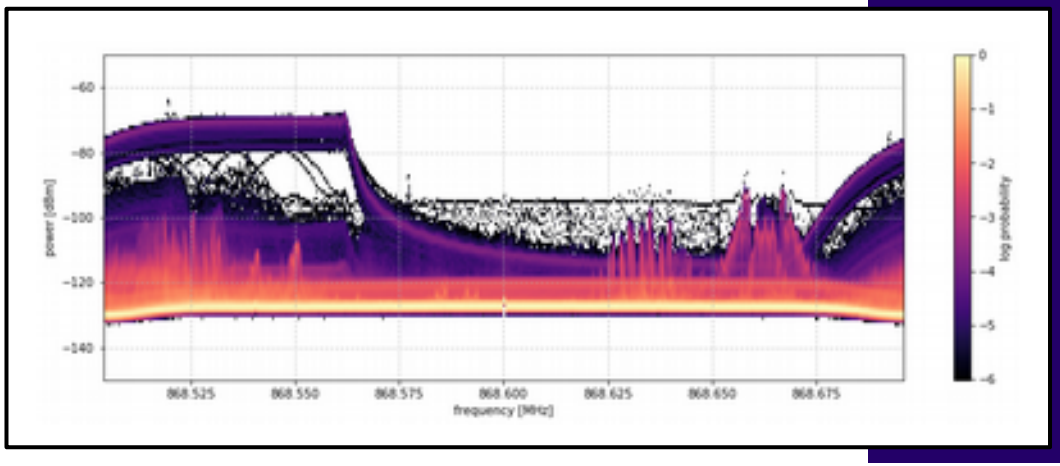
- We are near the LTE base station on 800 MHz band
  - Very strong in-band and out-of-band interference
- SNE-ESHTER data useless due to LNA saturation.
  - Antenna is directly in the beam of the LTE BS.
- SIGFOX moved uplink band to 868.600 MHz
  - apparently less in-band interference,
  - can also provide cavity band-pass filter
  - still need to check how much the sensitivity improved.

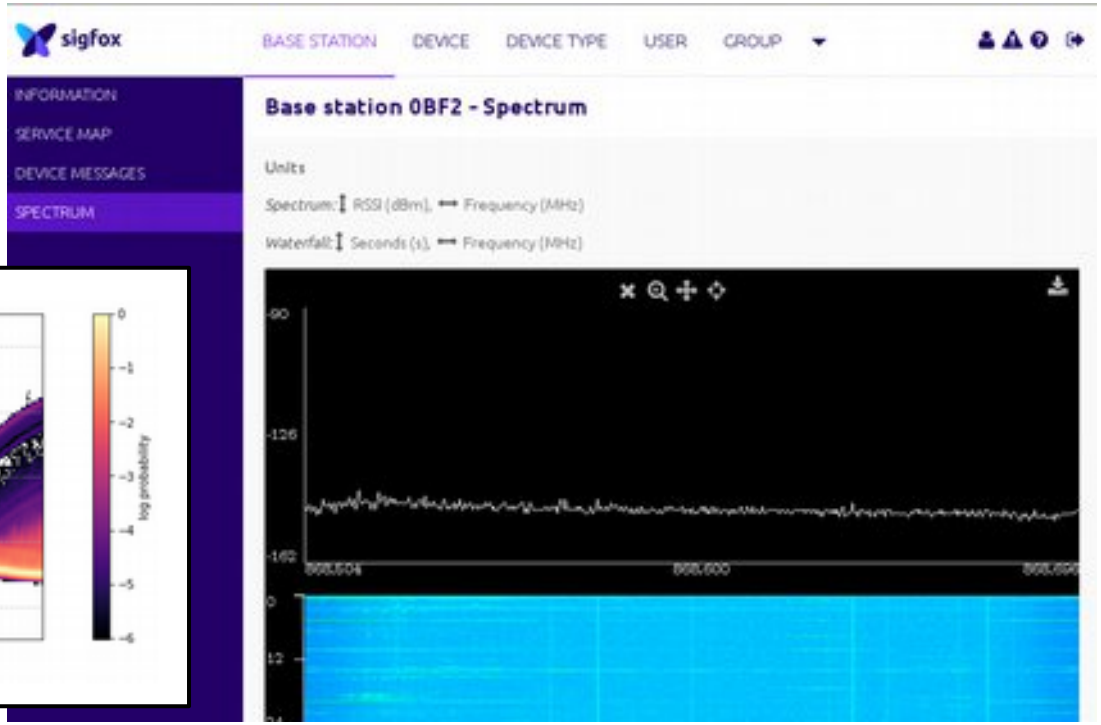
Improve packet datasets

- Collect more data:
  - Longer term, more packets, more locations, longer distance links ...
- Correct problem with low PRR
  - Devices with low PRR are automatically blocked from network by some mechanism in SIGFOX backend.
- Integrate LQE with physical layer sensing
  - Can LQE improve upon the results from spectrum sensing only?

Spectrum from the base station

- High-quality baseband data from SIGFOX backend
  - same antenna as receiver – more representative
  - much better sensitivity and lower phase noise
- Currently have
  ~ 6 days of data





Questions?

Tomaž Šolc
tomaz.solc@ijs.si