Synchronization in distributed SDR for localization applications

The challenge of nanosecond accuracy

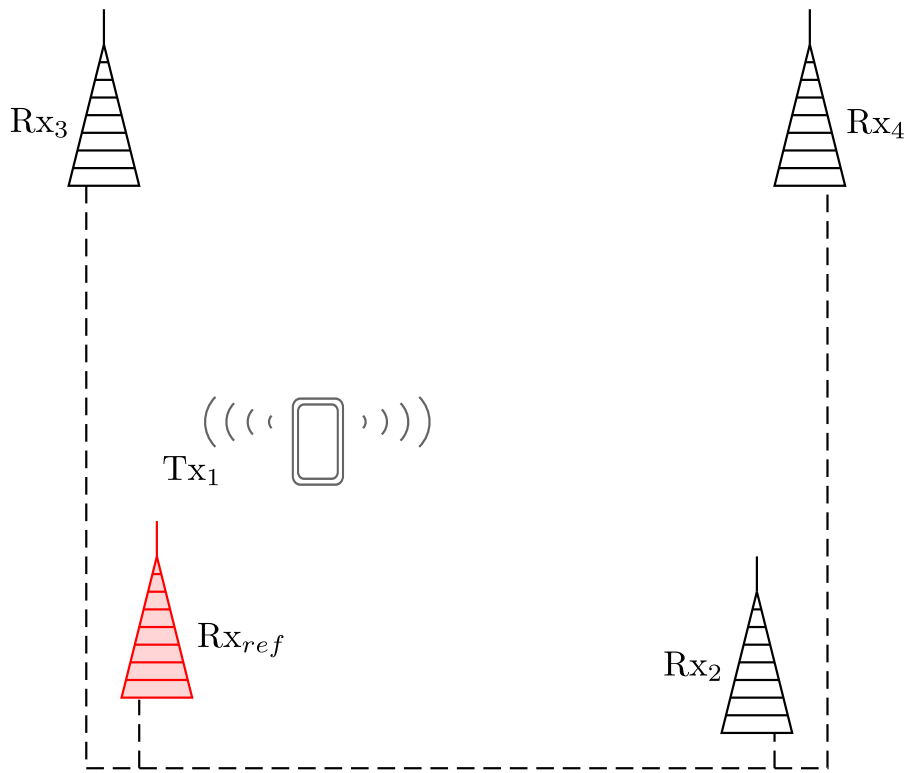
Johannes Schmitz, Manuel Hernández

January 31, 2016

Institute for Theoretical Information Technology
Prof. Dr. Rudolf Mathar
RWTH Aachen University

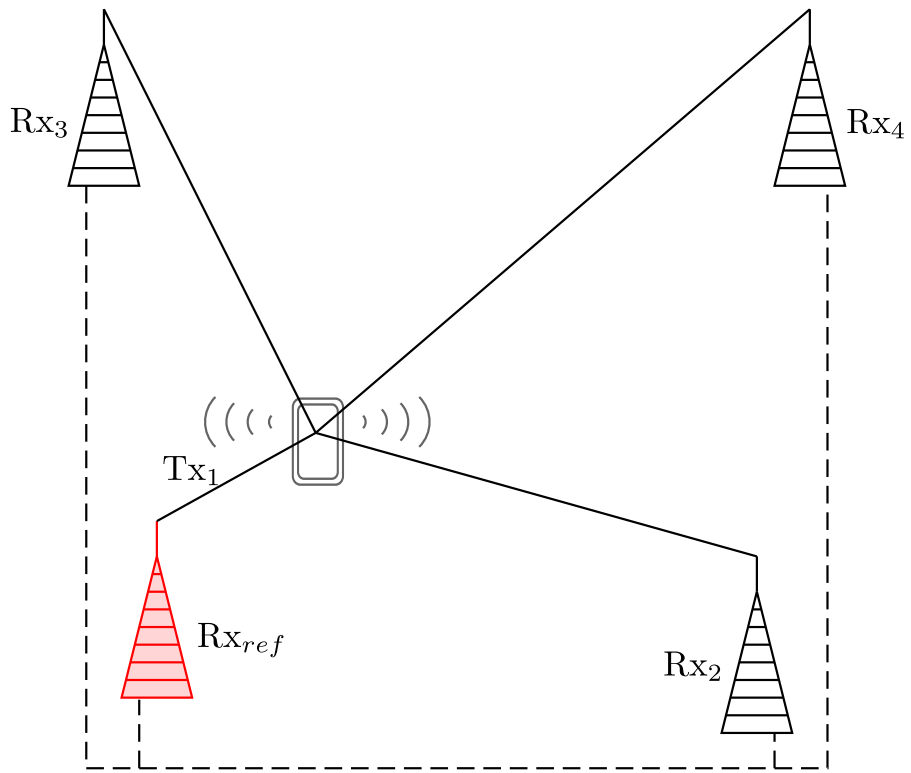
Introduction

- Time synchronized receivers (sensors, anchors, anchor nodes)
- Able to exchange samples
- Reference receiver (fusion center)
- Time difference of arrival (TDOA) measurements



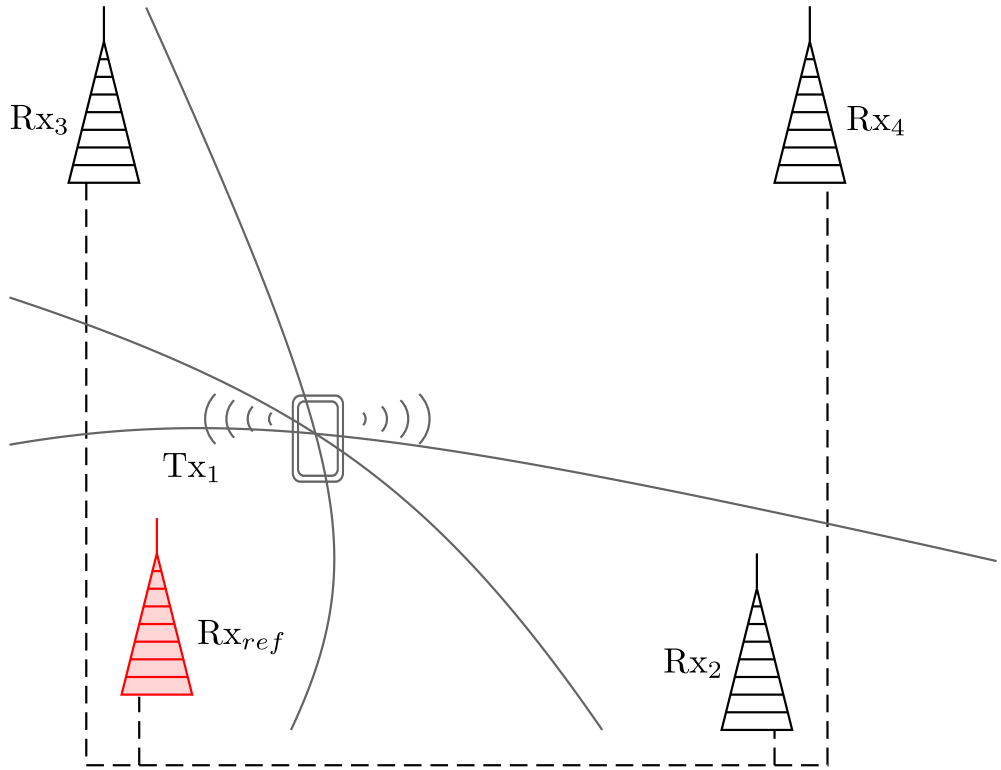
Introduction

- Time synchronized receivers (sensors, anchors, anchor nodes)
- Able to exchange samples
- Reference receiver (fusion center)
- Time difference of arrival (TDOA) measurements



Introduction

- Time synchronized receivers (sensors, anchors, anchor nodes)
- Able to exchange samples
- Reference receiver (fusion center)
- Time difference of arrival (TDOA) measurements

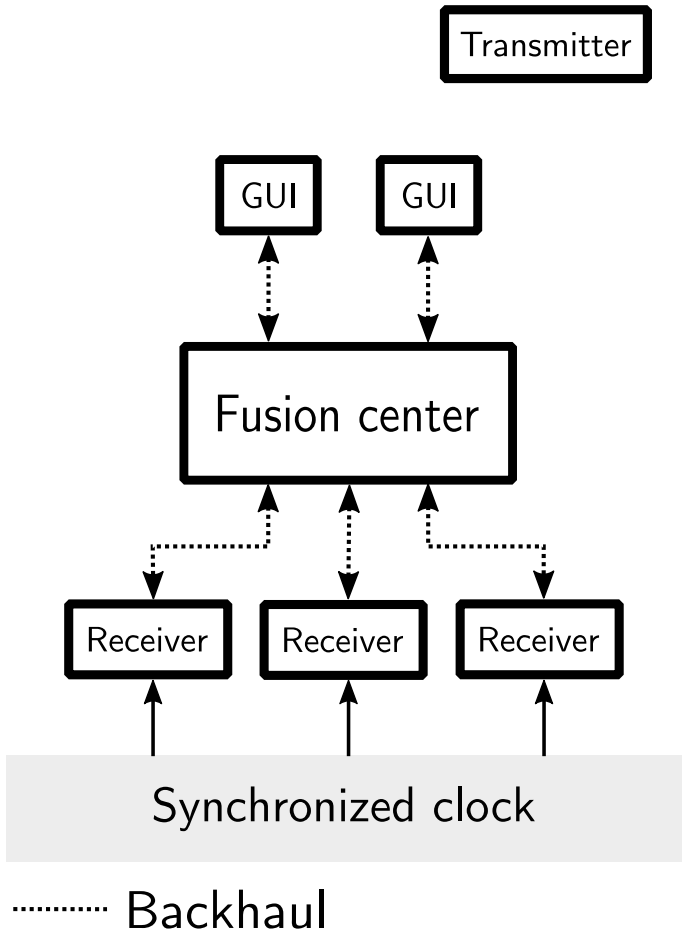


Introduction

- To our knowledge no existing open source SDR framework for real-time TODA based radio localization
- Its pretty tough mainly due to the speed of light
  - 1 ns equals 30 cm (one foot) of propagation!
- Many people have build ultrasound based systems
- Some ultra wideband systems exist
- Some people do signal recording and "offline" processing
- Commercial or military systems extremly expensive
- It's a distributed system
  - A lot of hardware, logistic problems, network programming

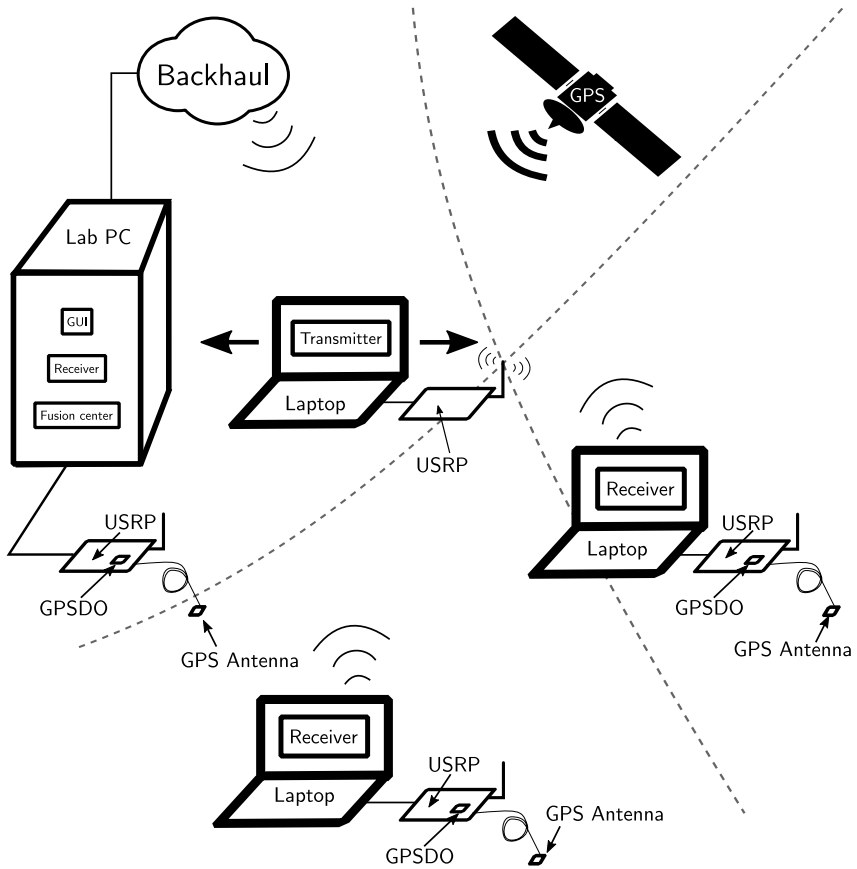
System architecture: Software layer

- Flexible architecture for different scenarios and algorithms
- Software components
  - GNU Radio
  - Python
  - Qt
  - ØMQ
- Nodes require a backhaul connection to communicate both samples and commands



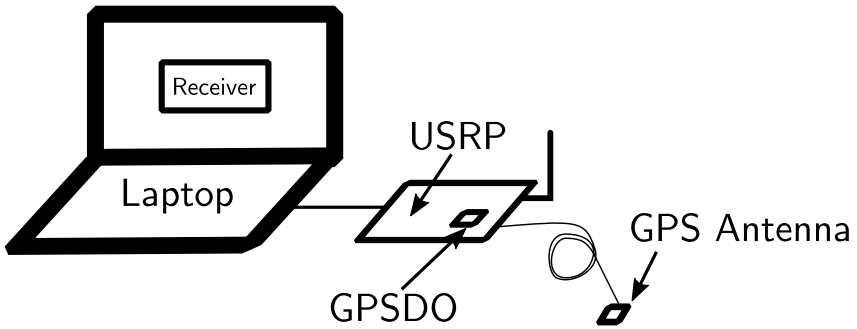
System architecture: Hardware layer

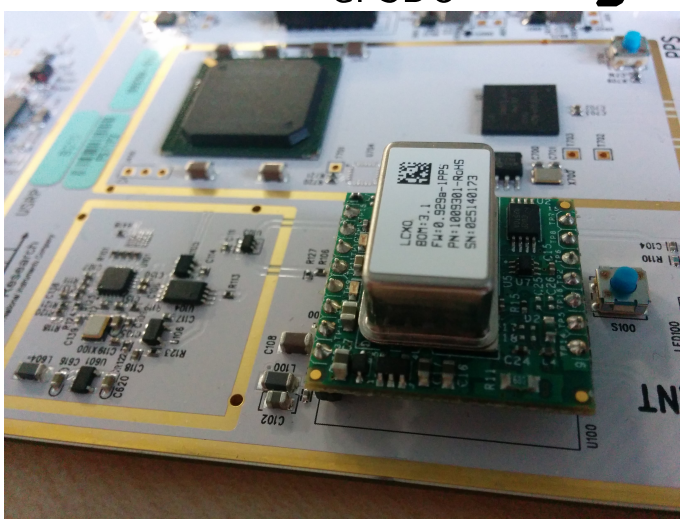
- Distributed system
- Variable number of nodes
- Real time results
- Compatibility with different GPS disciplined oscillators (GPSDOs)
  - Jackson Labs/Ettus LCXO
  - Jackson Labs LTE Lite
  - ...



System architecture: Hardware layer

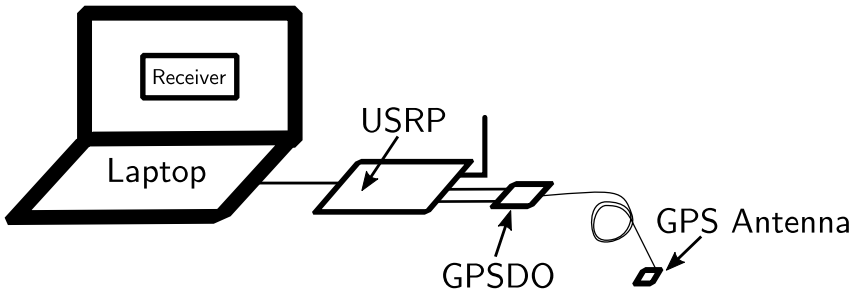
- Distributed system
- Variable number of nodes
- Real time results
- Compatibility with different GPS disciplined oscillators (GPSDOs)
  - Jackson Labs/Ettus LCXO
  - Jackson Labs LTE Lite
  - ...

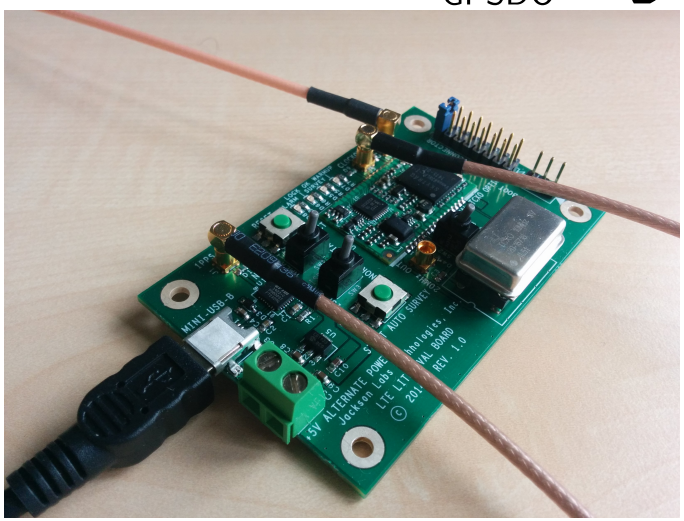




System architecture: Hardware layer

- Distributed system
- Variable number of nodes
- Real time results
- Compatibility with different GPS disciplined oscillators (GPSDOs)
  - Jackson Labs/Ettus LCXO
  - Jackson Labs LTE Lite
  - ...

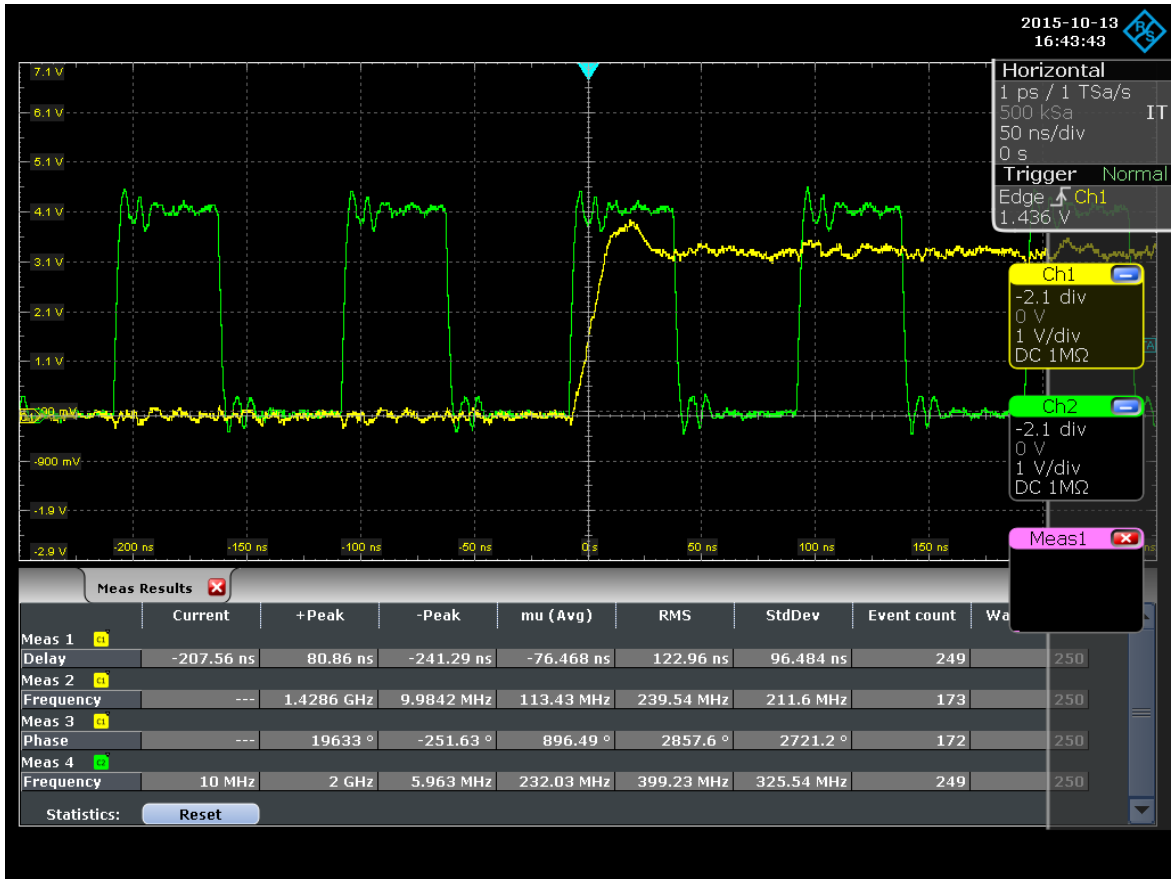




System architecture: Hardware layer

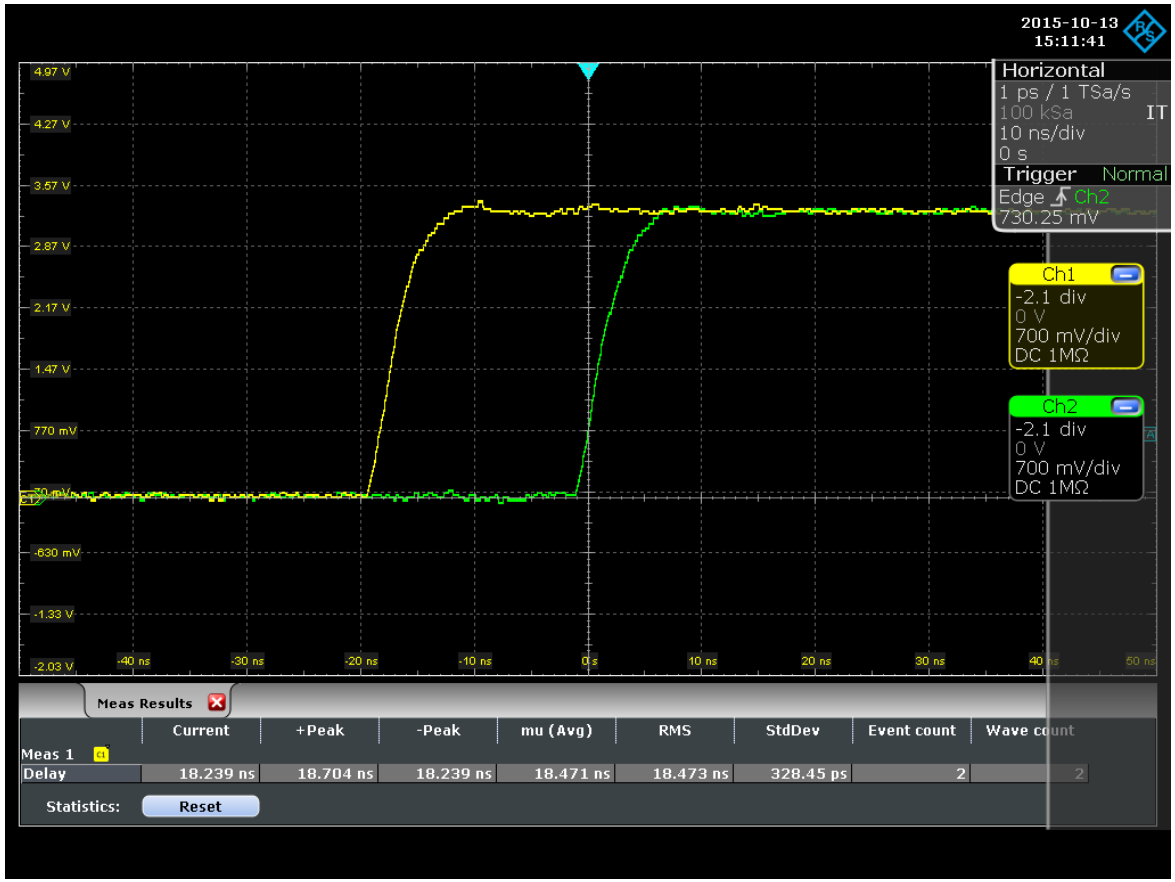
Timing synchronization

- Coordinated Universal Time (UTC)
- Pulse per second (PPS)
- 10MHz Clock
- Matching issues (50 Ohm)



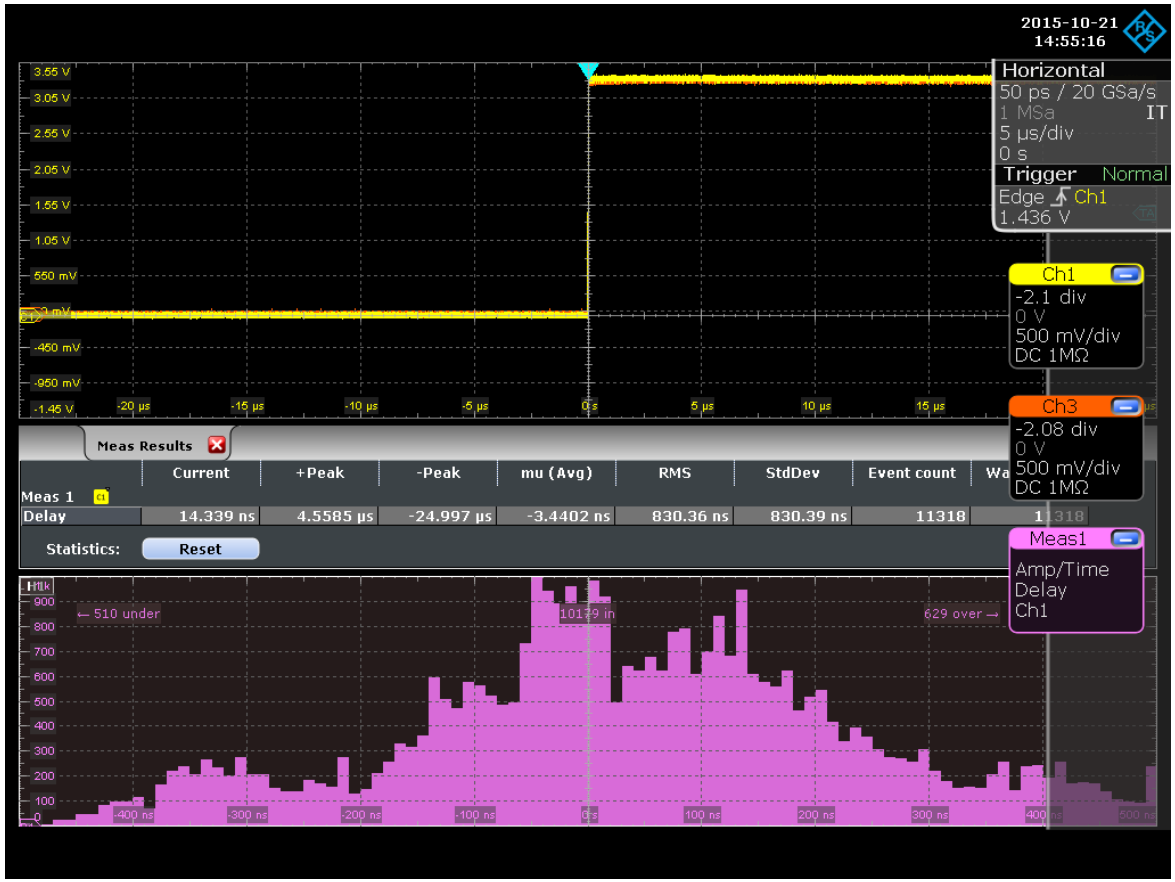
Timing synchronization

- Coordinated Universal Time (UTC)
- Pulse per second (PPS)
- 10MHz Clock
- Matching issues (50 Ohm)



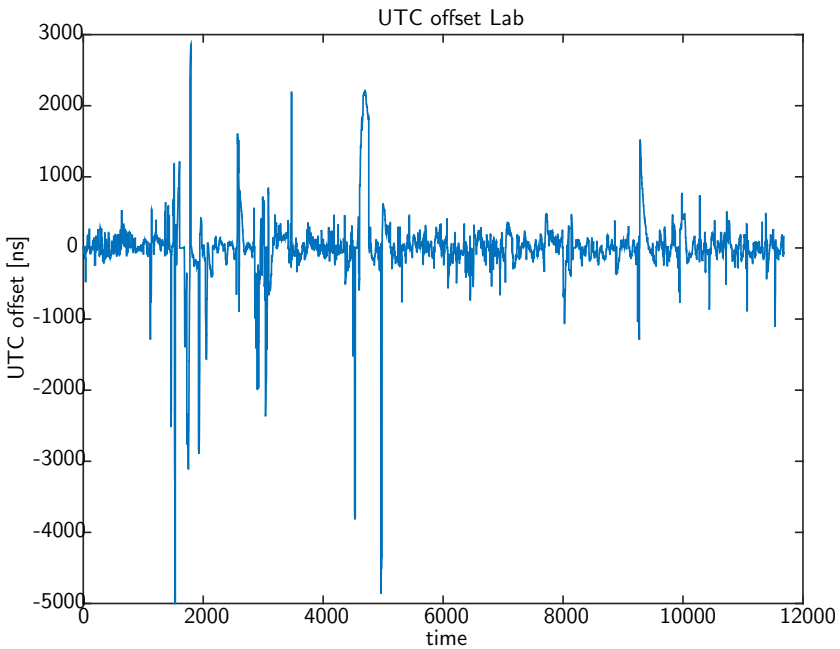
Timing synchronization

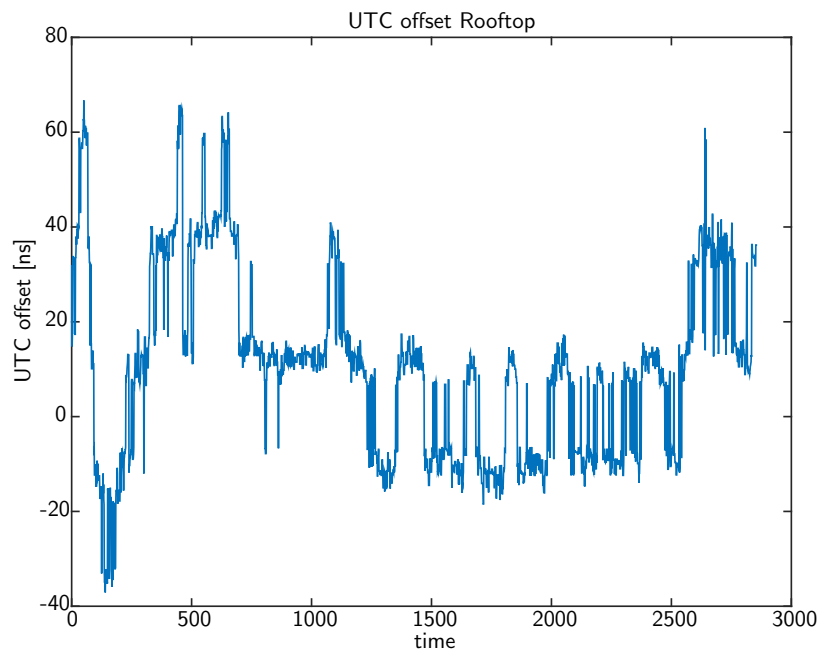
- Coordinated Universal Time (UTC)
- Pulse per second (PPS)
- 10MHz Clock
- Matching issues (50 Ohm)
- Can take one hour to have a good and stable fix



Results: LTE Lite reception comparison

- GPS reception big issue
- Outside window of lab peaks of thousands of ns
- On the rooftop stable within 50ns





UHD/GNU Radio API

- Very helpful:
  `http://files.ettus.com/manual/page_sync.html`
- in general relatively large delays in SDR systems!
  - need to synchronize the frontends for high accuracy

1. query the GPSDO for seconds and find PPS
2. now you have $\sim$ 1s to react before the next PPS
3. tell the device to set the internal time (+1s) on the next PPS
   - UHD/GNU Radio: `set_time_next_pps(...)`
4. use ntp for synchronization of the hosts

Issues and comments

- Ettus devices:
  - works well with UHD 3.8.5
  - issues (multichannel, synchronization) with 3.9 series
    - Wait for 3.10
    - Maybe some additional information in the manual/documentation for API changes (something changed according to changelog)
    - If necessary work with support to track down the bugs
    - Test cases in internal Ettus quality control for signal integrity along all channels?
  - Phase coherent synchronization is a different story

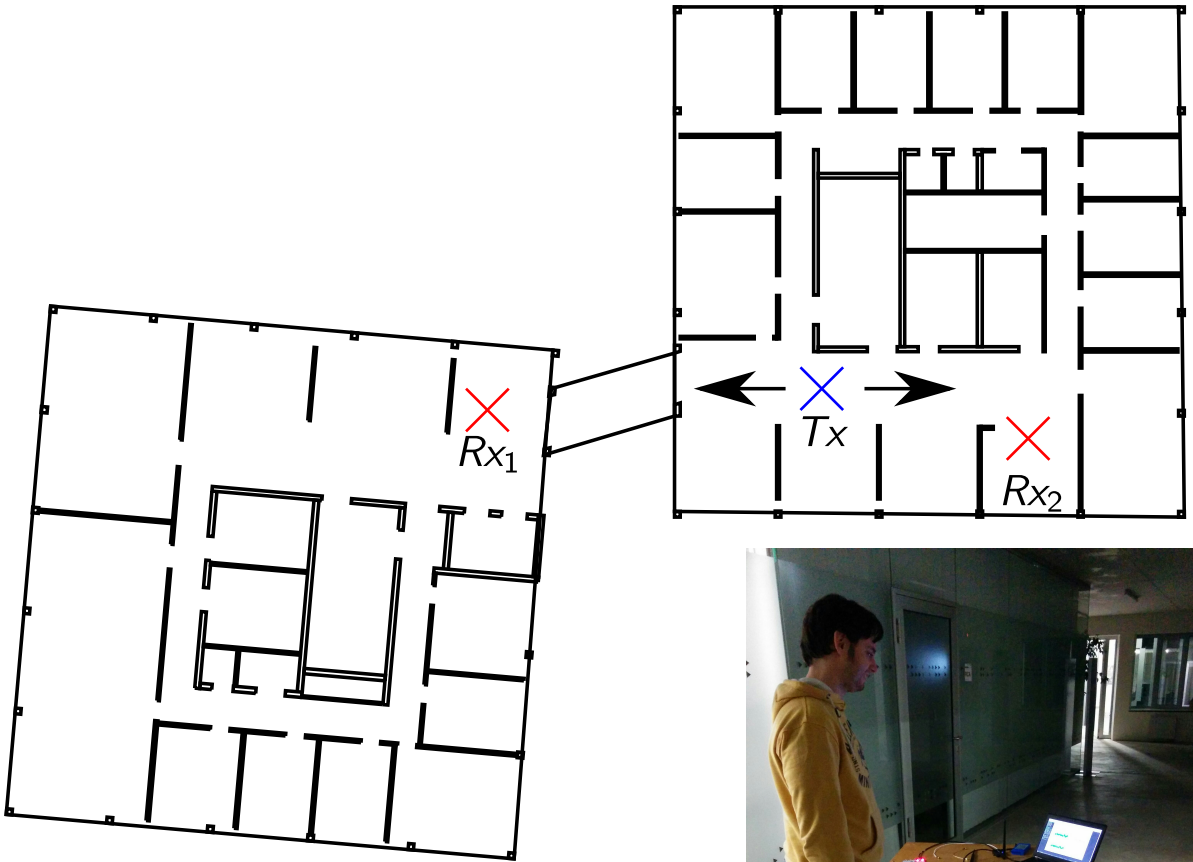
Network programming

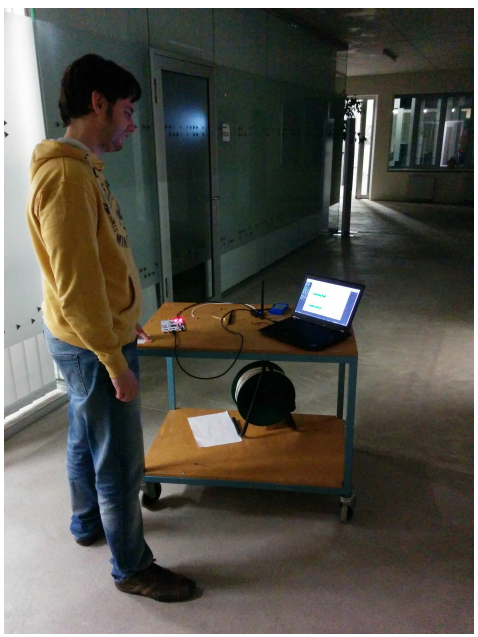
- Send message from fusion center to all receivers with time to receive and number of samples
  - use UHD/GNU Radio stream command API

```
stream_cmd =
   uhd.stream_cmd(uhd.stream_cmd_t.STREAM_MODE_NUM_SAMPS_AND_DONE)
stream_cmd.num_samps = samples_to_receive
stream_cmd.stream_now = False
stream_cmd.time_spec = time_to_sample
self.usrp_source.issue_stream_cmd(stream_cmd)
```

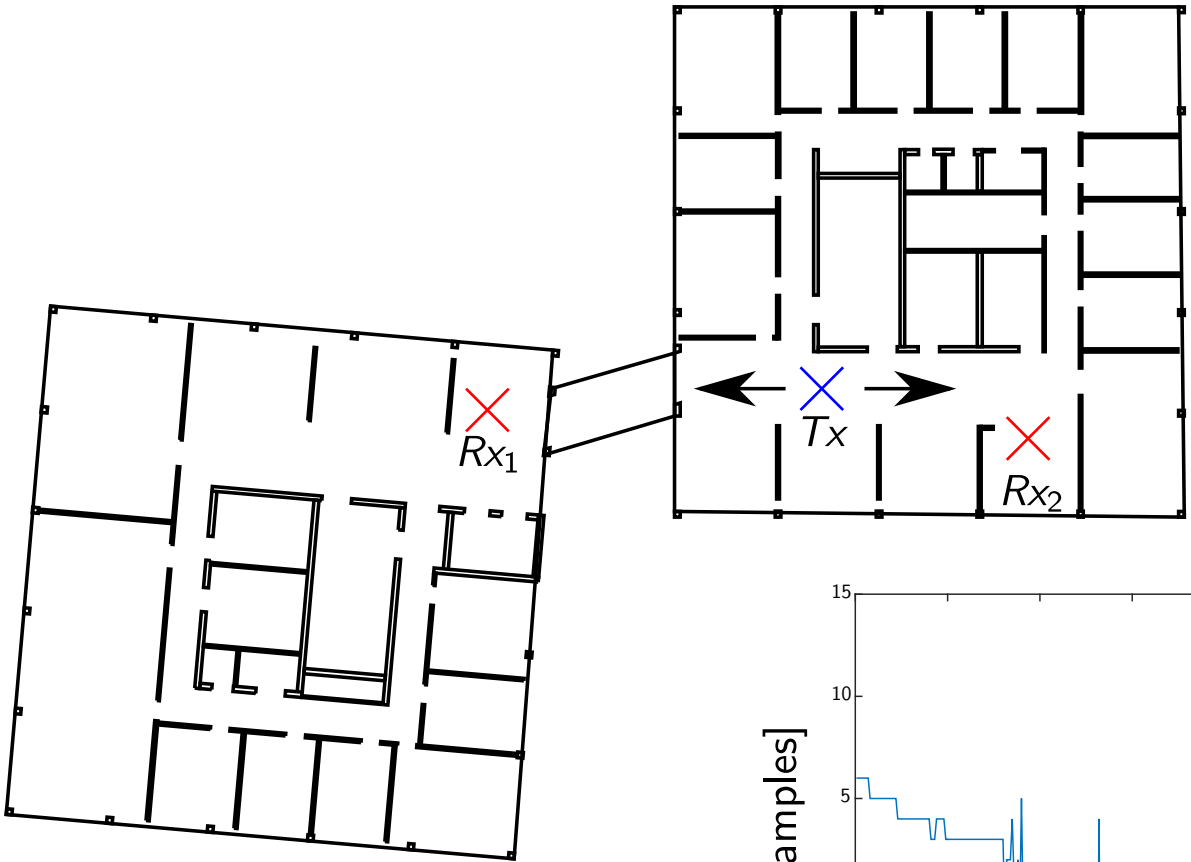
- Wait for the samples and process in the fusion center
- provide results to all GUIs
- We use GNU Radio zeromq blocks for this
- General problems: throughput limit of the backbone, e.g., WiFi

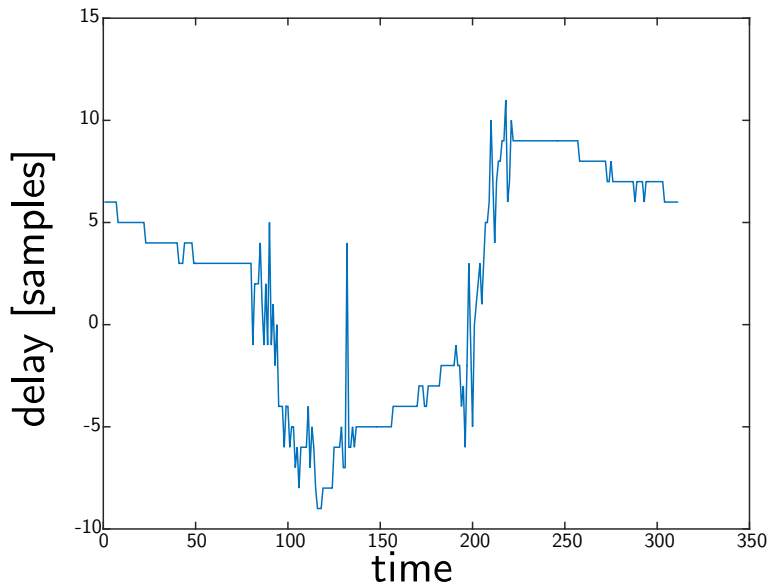
Results: Walk along the corridor



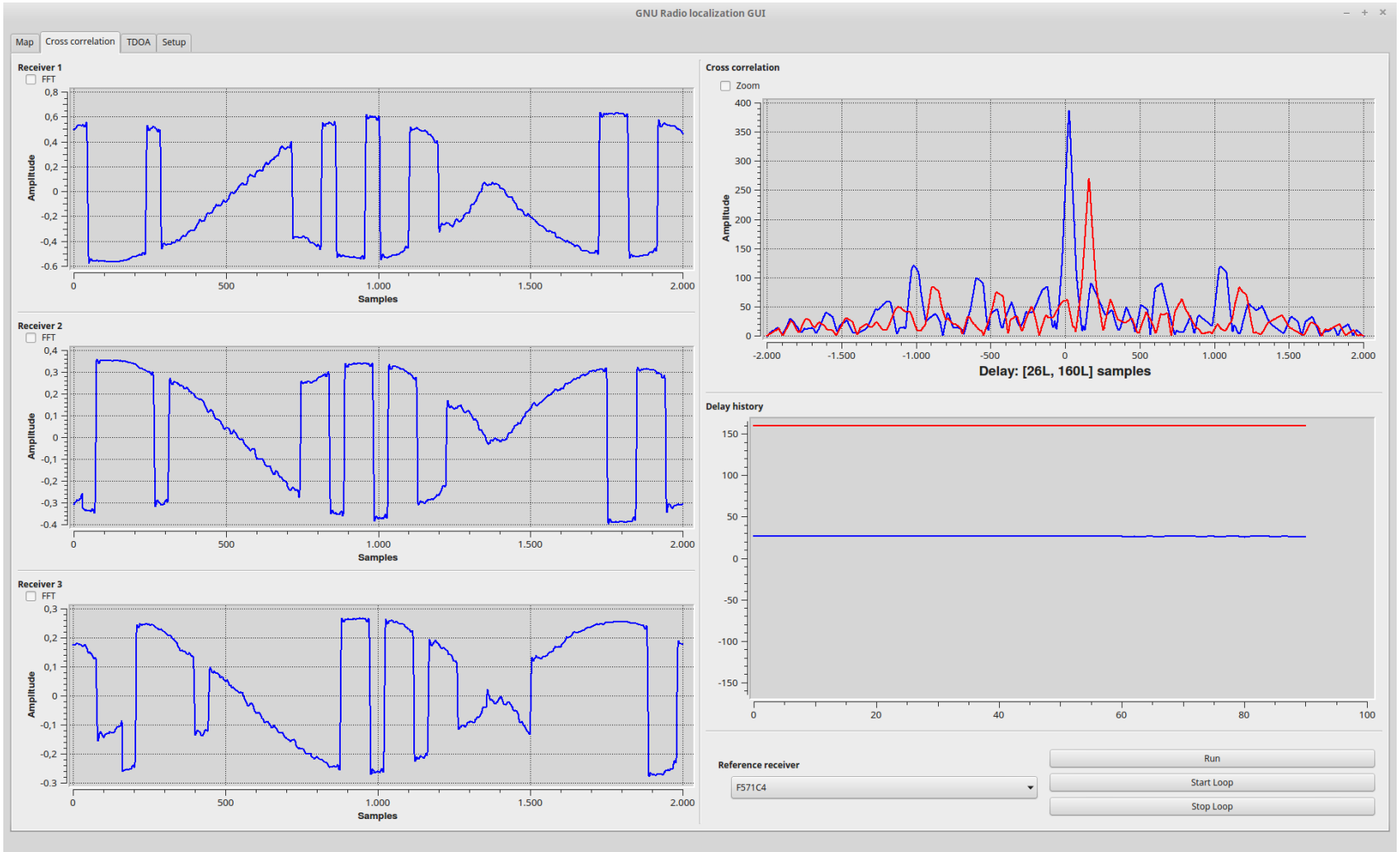


Results: Walk along the corridor

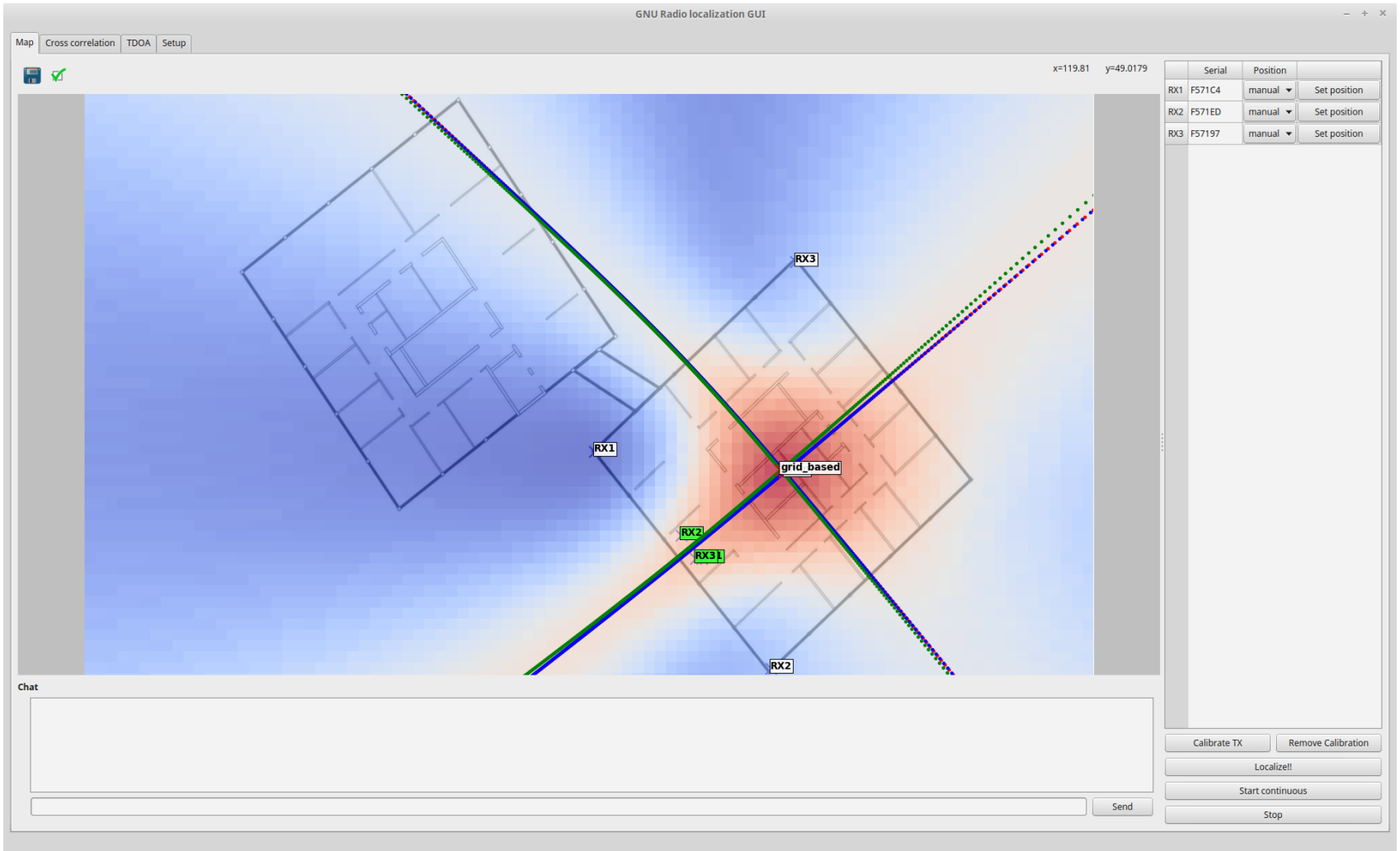




System architecture: GUI



System architecture: GUI



Final comments and outlook

- Timing improvement: use a GPSDO that is able to run in "1D-Mode" with fixed position
- Use a reference station with a known position to calibrate out the timing drift
  - problems with fast retuning of USRPs
  - need stream command type of API for tune requests
- Ideal solution: RTK (Differential GPS)
  - provide GPS raw data through UHD

Finish

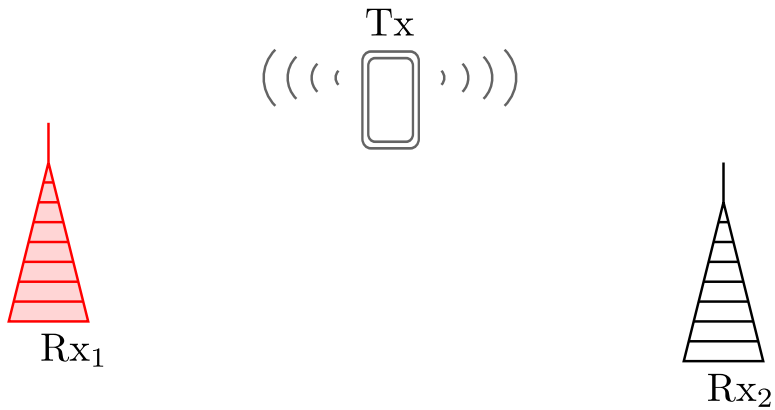
Thank You! Questions?

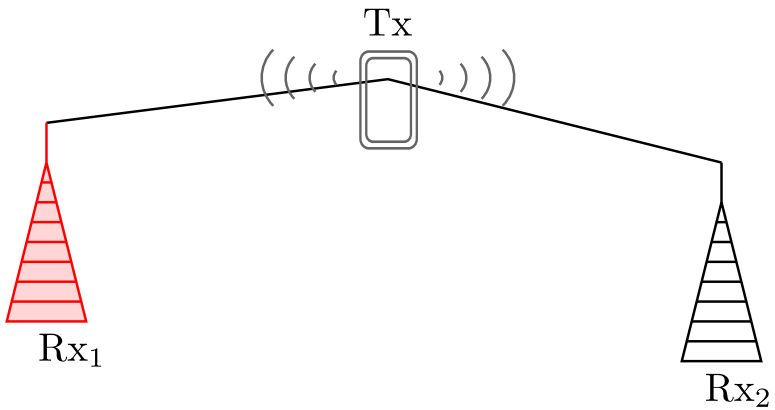
Finish

Backup slides

TDOA

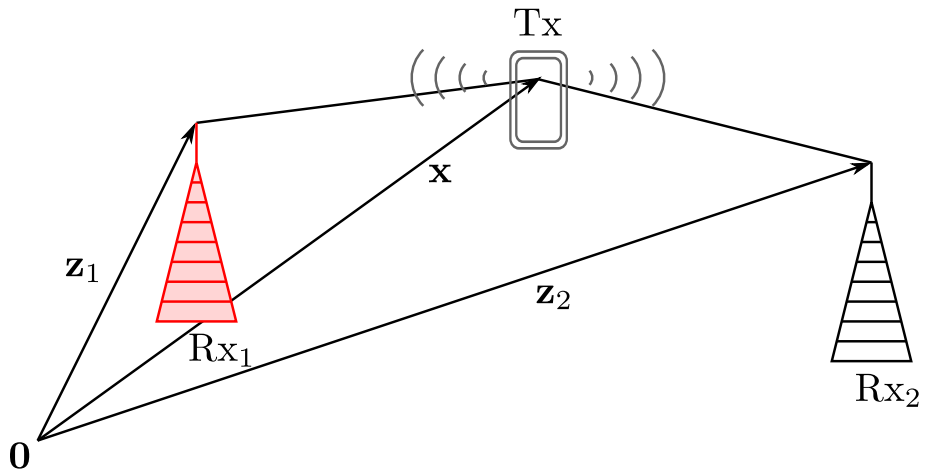
Recap





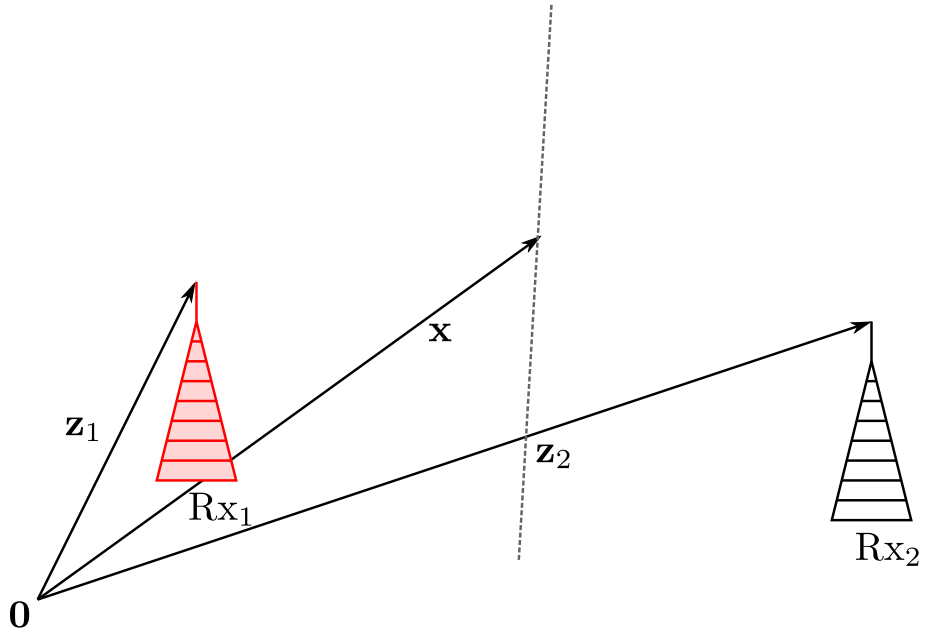
- No direct ranging possible, system limitation, e.g., non cooperative case

TDOA
Recap



- No direct ranging possible, system limitation, e.g., non cooperative case
- Use time **difference** of arrival (TDOA), c is the speed of the wave
  $\rightarrow \Delta(\mathbf{x}, \mathbf{z}_k, \mathbf{z}_l) = \frac{1}{c}\|\mathbf{z}_k - \mathbf{x}\|_2 - \frac{1}{c}\|\mathbf{z}_l - \mathbf{x}\|_2$

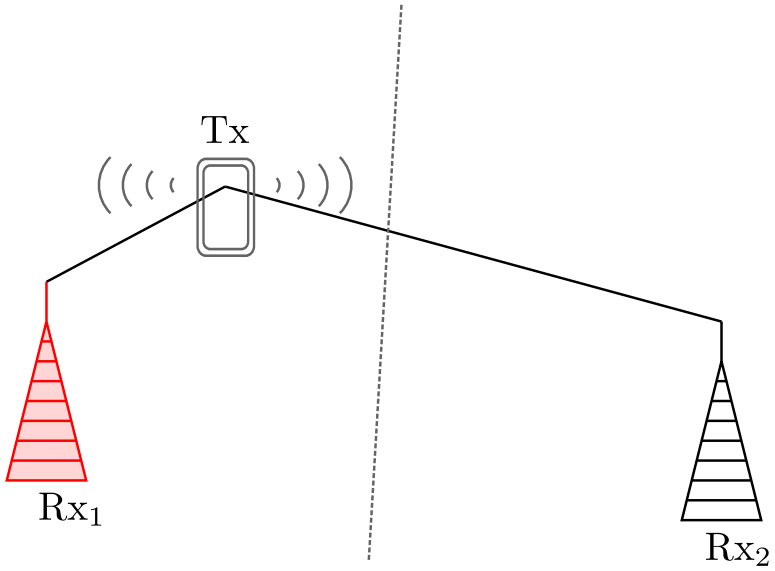
TDOA
Recap



- No direct ranging possible, system limitation, e.g., non cooperative case
- Use time **difference** of arrival (TDOA), c is the speed of the wave
  $\rightarrow \Delta(\mathbf{x}, \mathbf{z}_k, \mathbf{z}_l) = \frac{1}{c}\|\mathbf{z}_k - \mathbf{x}\|_2 - \frac{1}{c}\|\mathbf{z}_l - \mathbf{x}\|_2$

TDOA

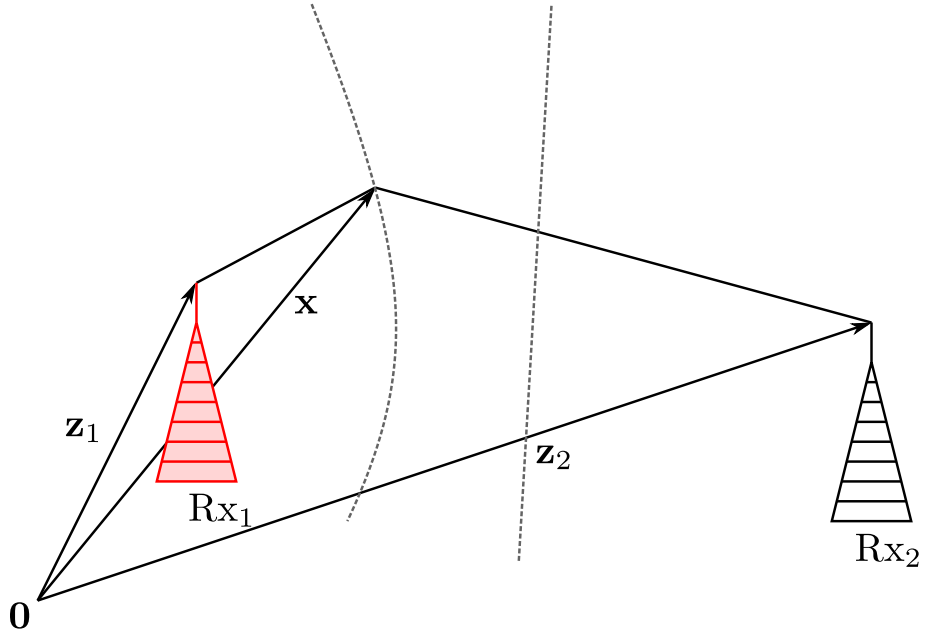
Recap



- No direct ranging possible, system limitation, e.g., non cooperative case
- Use time **difference** of arrival (TDOA), c is the speed of the wave
  $\rightarrow \Delta(\mathbf{x}, \mathbf{z}_k, \mathbf{z}_l) = \frac{1}{c}\|\mathbf{z}_k - \mathbf{x}\|_2 - \frac{1}{c}\|\mathbf{z}_l - \mathbf{x}\|_2$

TDOA

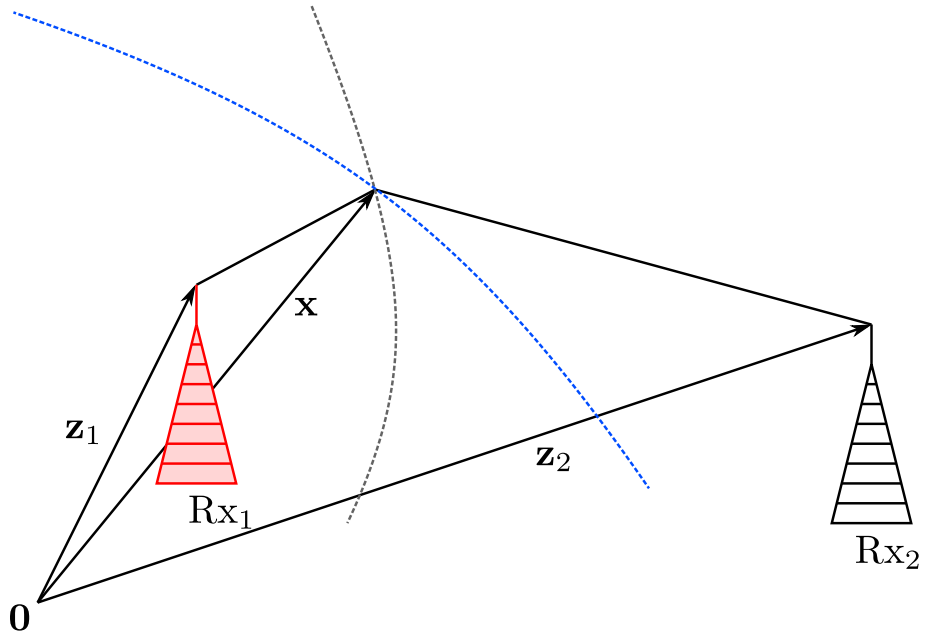
Recap



- No direct ranging possible, system limitation, e.g., non cooperative case
- Use time **difference** of arrival (TDOA), c is the speed of the wave
  $\rightarrow \Delta(\mathbf{x}, \mathbf{z}_k, \mathbf{z}_l) = \frac{1}{c}\|\mathbf{z}_k - \mathbf{x}\|_2 - \frac{1}{c}\|\mathbf{z}_l - \mathbf{x}\|_2$

TDOA

Recap



- No direct ranging possible, system limitation, e.g., non cooperative case
- Use time **difference** of arrival (TDOA), c is the speed of the wave
  $\rightarrow \Delta(\mathbf{x}, \mathbf{z}_k, \mathbf{z}_l) = \frac{1}{c}\|\mathbf{z}_k - \mathbf{x}\|_2 - \frac{1}{c}\|\mathbf{z}_l - \mathbf{x}\|_2$
- Well known classical algorithms, e.g., [CH94]
- Grid based algorithm [SDM15]

Bibliography

Y. T. Chan and K. C. Ho.
A simple and efficient estimator for hyperbolic location.
IEEE Trans. Signal Process., 42(8):1905–1915, Aug 1994.

J. Schmitz, D. Dorsch, and R. Mathar.
Compressed time difference of arrival based emitter localization.
In *Proc. 3rd Int. Workshop on Compressed Sensing Theory and its Applications to Radar, Sonar and Remote Sensing (CoSeRa 2015)*, pages 1–5, Pisa, Italy, June 2015.