NOKIA

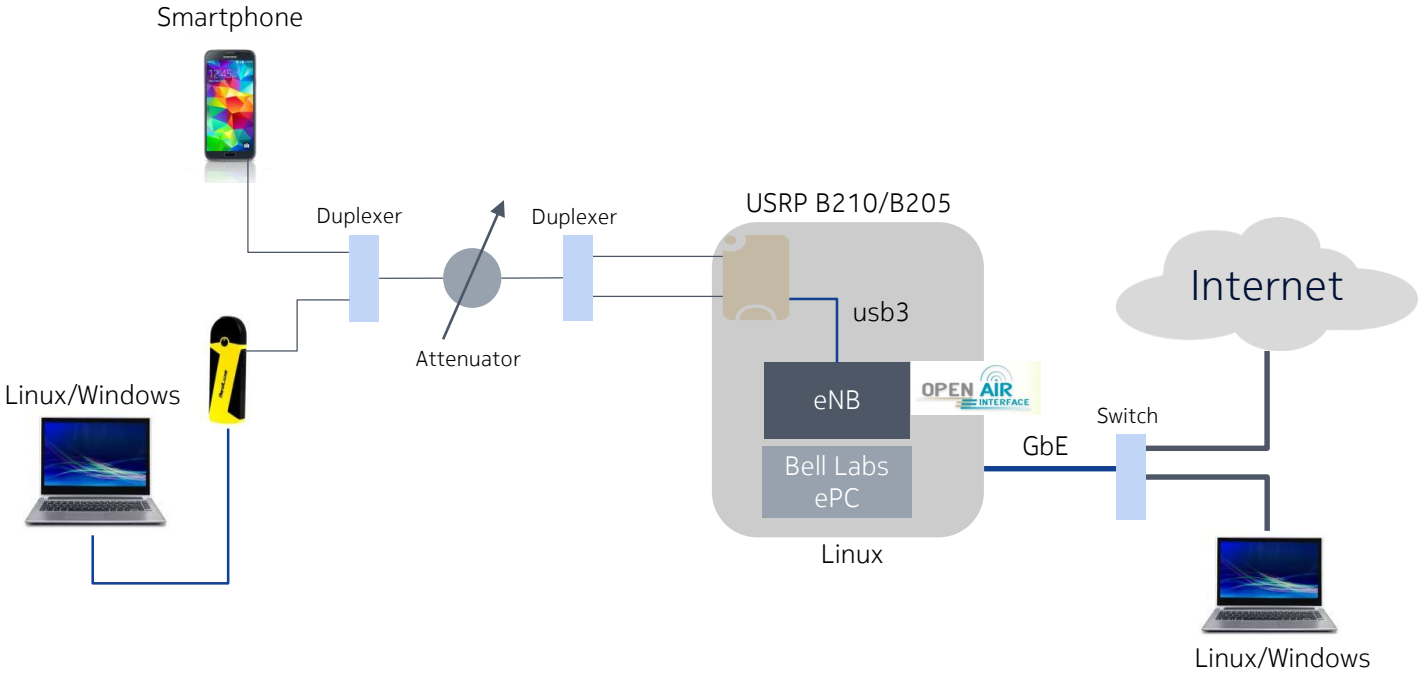
Bell Labs

I/Q Record-Playback in OAI

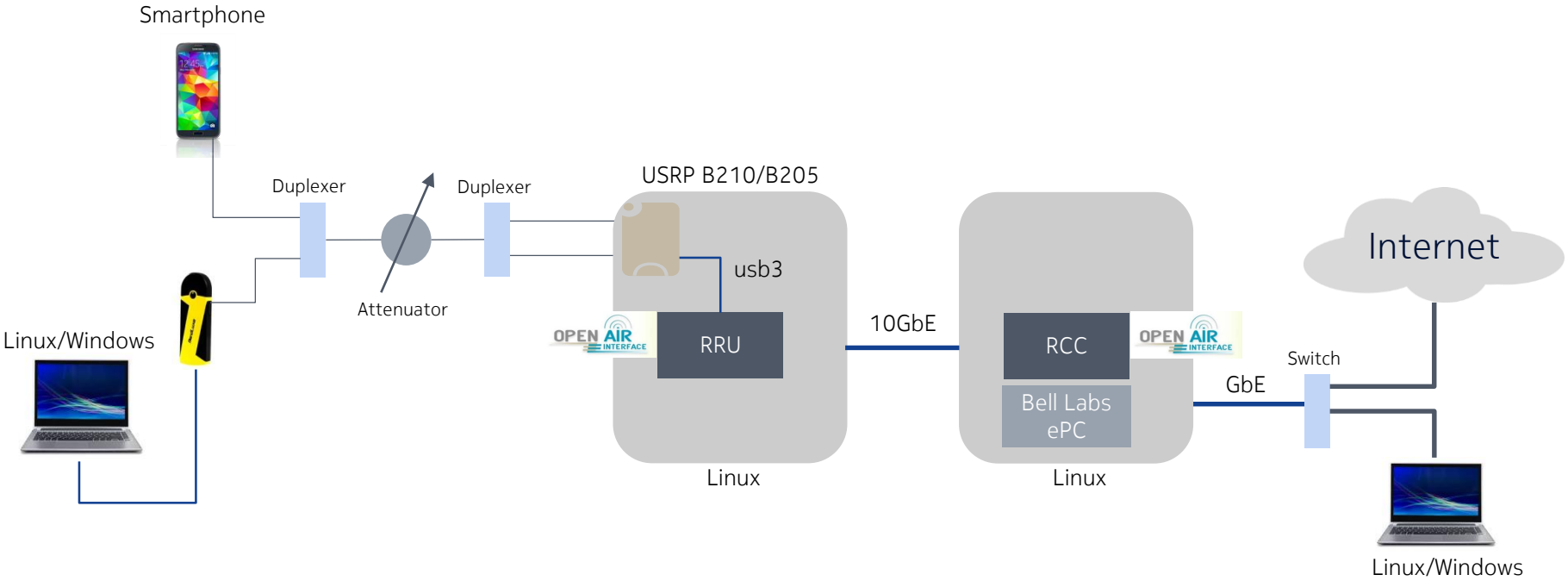
Bruno Mongazon-Cazavet, Erick Bizouarn

Tuesday 7 November 2017

OAI FULL ENB S1 SETUP @ BELL LABS



OAI SPLIT ENB S1 SETUP @ BELL LABS



PROBLEM STATEMENT

- eNB with radio board attached requires “periodic 1ms I/O”
- This makes the following operations very challenging/not possible:
  - **Step-by-step debugging**
    Not possible without breaking the radio operation.
  - **Memory consumption checking**
    Not possible because instrumented code executes too slowly.
  - **Source code Coverage analysis, learning and rework**
    Not possible because instrumented/optimized code executes too slowly.
  - **Non-regression testing and features validation**
    Testing with real radio is time and resources consuming, hard to reproduce with accuracy.

CURRENT SOLUTIONS

- eNB diagnostic tools (LOG, T-Trace, softscope, gtk analysis, time measurements) are useful but:
  - **Instrumentation in the code disturbs code being instrumented**
    Heisenberg principle
  - **Instrumentation shall be asserted**
    Instrumentation is also code and not bug-free!
  - **Visual tools automation is painful**
    These tools should be avoided
- Current solutions do not allow incremental development life-cycle and slow bug-fix, features development and overall code quality. This is very critical from OSA collaborative point of view.

PROPOSED NEW SOLUTION

- **Record** “radio-activity” (subframes) in eNB at the lowest level in well-defined conditions and collect the respective initial (before recording) and final (after recording) external systems states
- Put external systems in their respective initial state and **replay** recorded radio-activity (subframes) in eNB at the lowest level. Collect the respective final (after replaying) external systems states
- Respective final states of external systems after recording and after replaying **shall be identical**.
- Make the solution agnostic to 1ms I/O, practical and automatable

ASSUMPTIONS

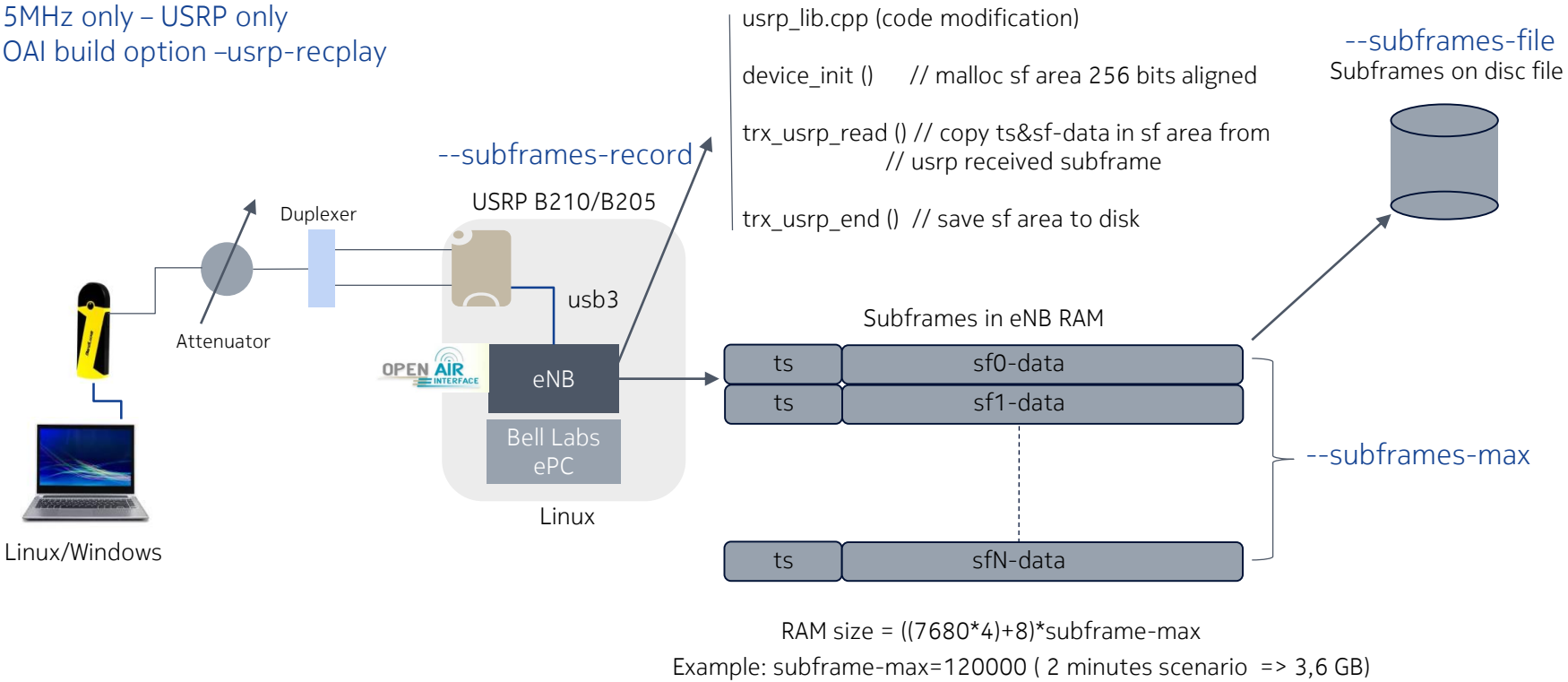
- Well-defined radio conditions
  - Can be arbitrary
  - Part of the record/playback scenario
- Respective states of external systems
  - Can be arbitrary
  - Part of the record/playback scenario
  - External systems are proofed (only the eNB is a System Under Test) and restart-able (can be put in their respective initial states)
  - External systems provide asserted/proofed state exhibition methods (so to allow for comparison methods)
- Can LTE be fully deterministic ?
  - Target an implementation with a deterministic behavior…

CAN LTE BE FULLY DETERMINISTIC ?

- C-RNTI allocation
  - OAI eNB currently allocates C-RNTI (0x003C-0xFFF3) in a random manner
  - The C-RNTI allocation is therefore modified to allocate a free value within a predefined C-RNTI range sized for the maximum number of UE's supported by the eNB.
- UE Authentication
  - The 3GPP standards enforces UE authentication through HSS system and involves UE and MME through a set of authentication control
  - The eNB code performing MAC authentication control is bypassed
  - Bell Labs ePC (ltebox) is modified to behave in 'bypass mode' regarding authentication
- Anything else ? Discussion about FFT/iFFT approximation...

RECORDING I/Q

5MHz only – USRP only
OAI build option –usrp-recplay



RAM size = ((7680*4)+8)*subframe-max
Example: subframe-max=120000 (2 minutes scenario => 3,6 GB)

REPLAYING I/Q

5MHz only – USRP only
OAI build option –usrp-replay

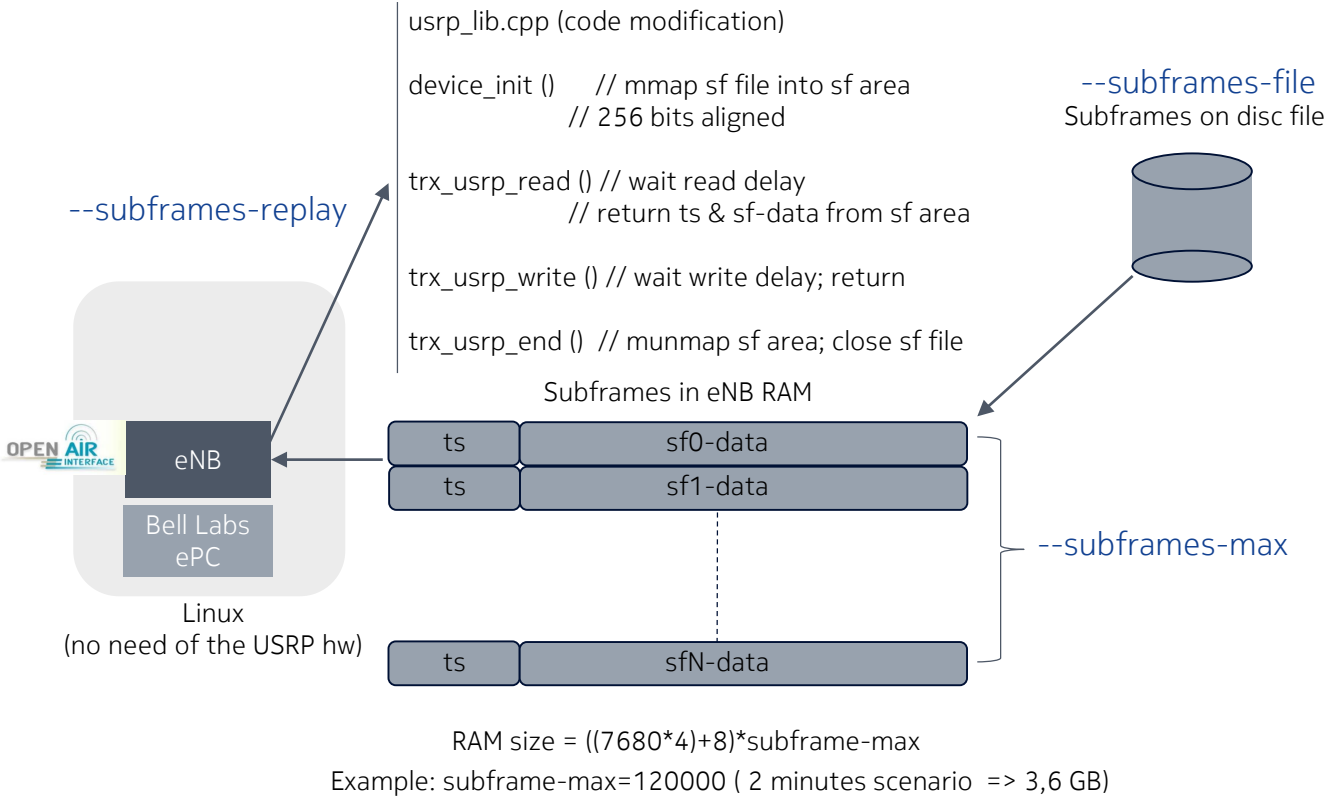
--subframes-read-delay
nb of µs to complete read

800µs matches real radio
slow down or accelerate

--subframes-write-delay
nb of µs to complete write

25µs matches real radio
slow down or accelerate

--subframes-loops
nb of loops to run the whole subframe area



ASSERTING CORRECTNESS

- **Record phase - Initial state**
  - UE: not connected
  - HSS: freshly started
  - ePC: freshly started
- **Record phase - scenario**
  - UE attaches
  - UE data traffic
  - UE detaches
- **Record phase - Final state**
  - UE: disconnected
  - HSS: has attached one UE
  - ePC: has attached/traffic/detached one UE
    counters.sh > final_state_record

- **Replay phase - Initial state**
  - UE: not connected
  - HSS: freshly started
  - ePC: freshly started
- **Replay phase - scenario**
  - UE attaches
  - UE data traffic
  - UE detaches
- **Replay phase - Final state**
  - UE: disconnected
  - HSS: has attached one UE
  - ePC: has attached/traffic/detached one UE
    counters.sh > final_state_replay

diff final_state_record final_state_replay ?

CURRENT USAGE

- **Replay I/Q files on full and split eNB**
  Works the same. For split eNB, compile and run RRU in replay mode, compile RCC in replay mode. Read/write delay can be used to artificially increase/decrease fronthaul bit rates.

- **valgrind**
  Some memory leaks found, to be fixed

- **kcachegrind**
  Code coverage

- **Test of re-architected code (RU-RAU-split branch)**

- **Debugging**

LIMITATIONS

- **5MHz only**
  subframe area limited to 5MHz – no global file information stored – can be easily improved. However 20MHz will consume a lot of RAM and disk space.

- **USRP only**
  USRP code modified. Might be re-architected (record/replay outside device ?)

- **UE data traffic shall be restartable**
  Use local (Bell Labs) ePC as destination IP is ok. Internet traffic is not necessarily restartable.

- **Timestamp overwrite with current time** (to be done soon)

REPLAY DEMO

- **I/Q files**
  A set of I/Q files have been recorded by Bell Labs:
  - 1 UE (attach, traffic, detach)
  - 2 UE's in parallel (attach, traffic, detach)

- **Local PC with OAI eNB built in record/replay mode**
  Without the need of radio board !

- **Replay one loop of I/Q with OAI code compiled with gcov**

- **Generate code coverage view (lcov)**

- **Look at OAI code coverage**


NOKIA