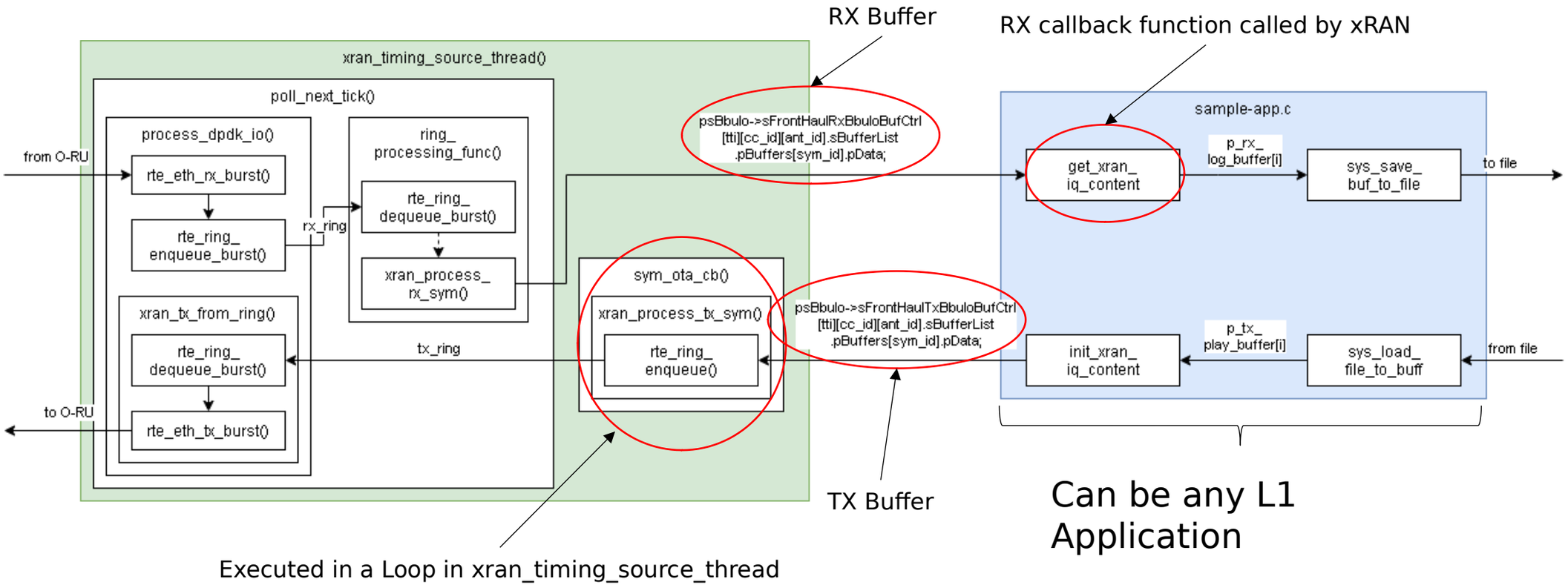
ORAN FHI library Integration

1. Library Understanding
2. Integration in OAI
3. Current Status

ORAN FHI Library explained

xRAN put RX data in the RX buffer, calls the RX callback function every
and calls the PRACH and SRS callback functions on every slot



xRAN reads the buffer linearly with time and enqueue the data it fo
in the TX ring so that it will be sent

RX callback function

- Synopsis:

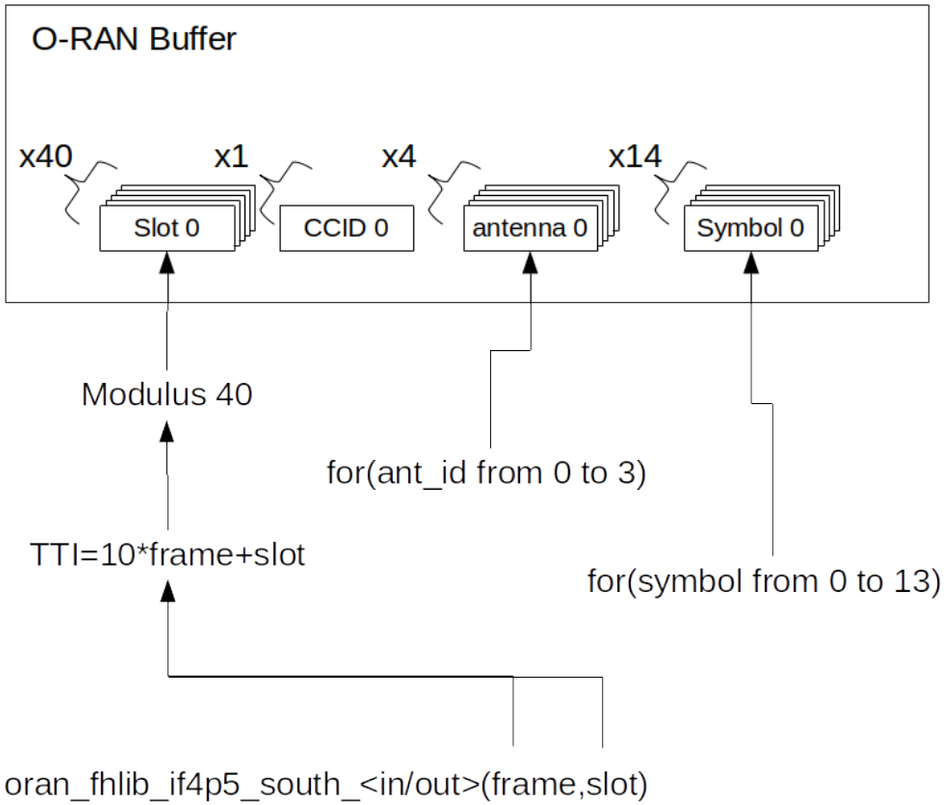
*void xran_fh_rx_callback(void *pCallbackTag, xran_status_t status) ;*

- Where:
  - *pCallbackTag* is a pointer to a structure *struct xran_cb_tag*.
  - *status* is a 32-bit integer equal to 0 if everything is normal.
- When:
  - On each half of a slot
  - On each end of a slot
- pCallback Tag contains the Timing information (TTI and Symbol) and the Sector identifier.

TX: Writing to the shared buffer

- Two buffers for two mappings of data:
  - Data Buffer mapped on TTI/symbol.
  - PRB Map buffer mapped on Section/PRB.
  - See: https://hackmd.io/@beurdouc/Hk5CVHM_Y?view#Operation-on-the-PRB-Element-and-the-Section-Descriptor

Timing Mapping in the Data Buffer



Timing

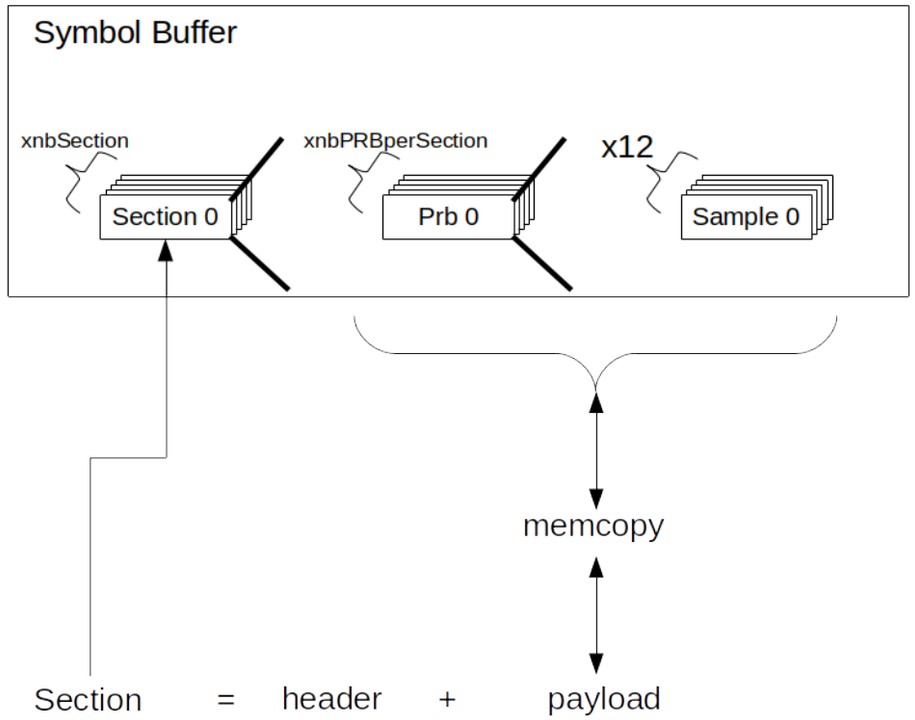
- The ORAN FHI library works with Over The Air time (time at the antenna).
- Within one second, the ORAN FHI library maintains the index of the current symbol *xran_lib_ota_sym_idx*.
- From this, it can determine the slot *xran_lib_ota_tti* and the index of the current symbol in the slot *xran_lib_ota_sym*.

Common Radio Protocol Mapping

Section Type 1,3 : DL/UL IQ data msgs									
0 (msb)	1	2	3	4	5	6	7 (lsb)	# of bytes	
transport header, see section 3.1.3								8	Octet 1
dataDirection	payloadVersion			filterIndex				1	Octet 9
frameId								1	Octet 10
subframeId				slotId				1	Octet 11
slotId		symbolid						1	Octet 12
sectionId								1	Octet 13
sectionId				rb	symInc	startPrbu		1	Octet 14
startPrbu								1	Octet 15
numPrbu								1	Octet 16
udCompHdr (not always present)								0/1	Octet 17
reserved (not always present)								0/1	Octet 18
udCompLen (not always present)								0/2	Octet 17/19
udCompParam (not always present)								0/1/2	Octet 17/19/21
iSample (1st RE in the PRB)								1*	K= 17/19/20/21/23
qSample (1st RE in the PRB)								1*	K+1*
…									
iSample (12th RE in the PRB)								1*	K+22*
qSample (12st RE in the PRB)								1*	K+23*
udCompParam (not always present)								0/1/2	K+24*
iSample (1st RE in the PRB)								1*	K+24/25/26*
qSample (1st RE in the PRB)								1*	K+25/26/27*
…									
iSample (12th RE in the PRB)								1*	K+46/47/48*
qSample (12st RE in the PRB)								1*	K+47/48/49*
…									
sectionId								1	Octet M
sectionId				rb	symInc	startPrbu		1	M+1
startPrbu								1	M+2
numPrbu								1	M+3
udCompHdr (not always present)								0/1	M+4
reserved (not always present)								0/1	M+5
udCompLen (not always present)								0/2	M+4/6
udCompParam (not always present)								0/1/2	M+4/6/8
iSample (1st RE in the PRB)								1*	K=M+4/6/7/8/10
qSample (1st RE in the PRB)								1*	K+1*
…									
iSample (12th RE in the PRB)								1*	K+22*
qSample (12st RE in the PRB)								1*	K+23*
udCompParam (not always present)								0/1/2	K+24*
iSample (1st RE in the PRB)								1*	K+24/25/26*
qSample (1st RE in the PRB)								1*	K+25/26/27*
…									
iSample (12th RE in the PRB)								1*	K+46/47/48*
qSample (12st RE in the PRB)								1*	K+47/48/49*

=

>



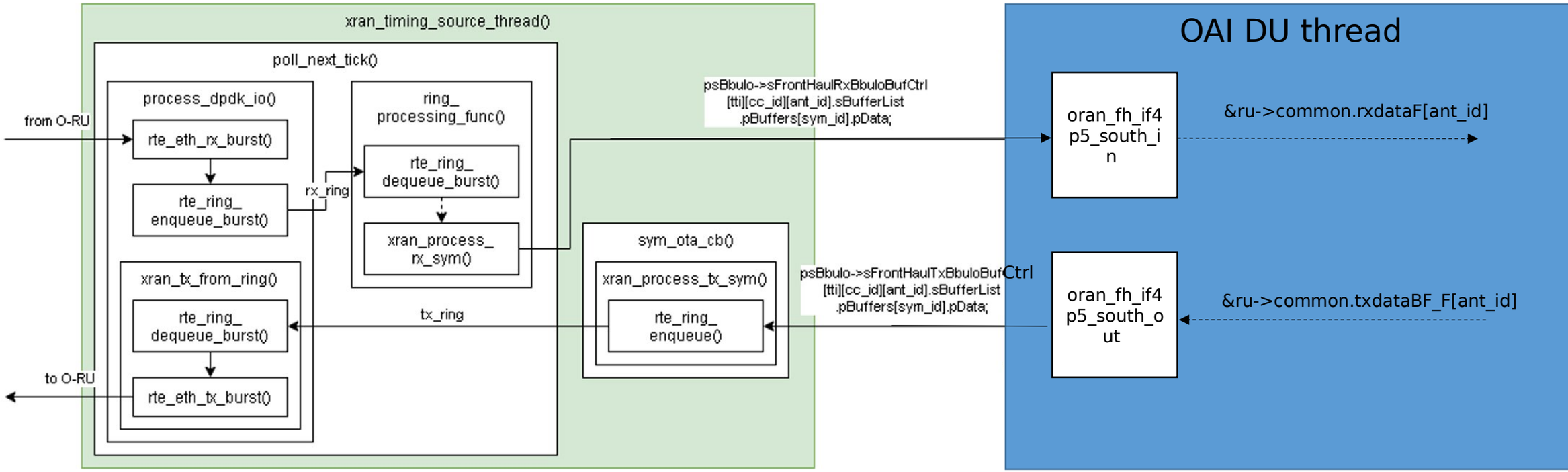
Assessing our understanding of the API: wrapper.cpp

- A simple test L1 application
- Relies on the *xran_lib_wrap* C++ header.
- Structure:
  - A callback function for U-Plane RX: *xran_fh_rx_callback*
  - A *main* function that:
    - Performs the initialization of the FHI.
    - Runs in an infinite loop to maintain the process alive.
- Reproduce the behavior of the sample app by sending content from files

Integration in OAI

- Junction with a wrapper library: liboran.so
- Implements the O-RAN 7.2 split interface
  - Common interface defined in:
    Targets/ARCH/COMMON/common_lib.h
  - Specific interface made of the two functions:
    - *oran_fh_if4p5_south_in* and *oran_fh_if4p5_south_out*
- Reuses the procedures tested in `wrapper.cpp`

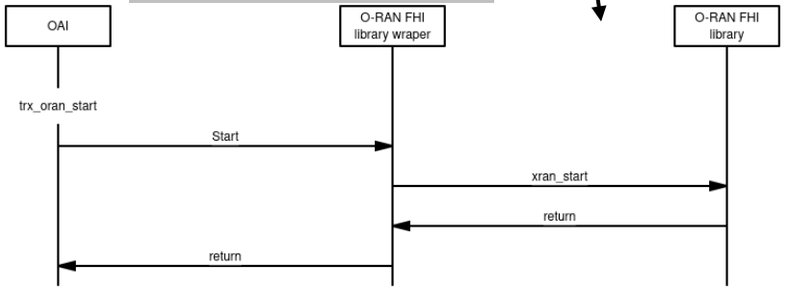
Wrapper library Architecture

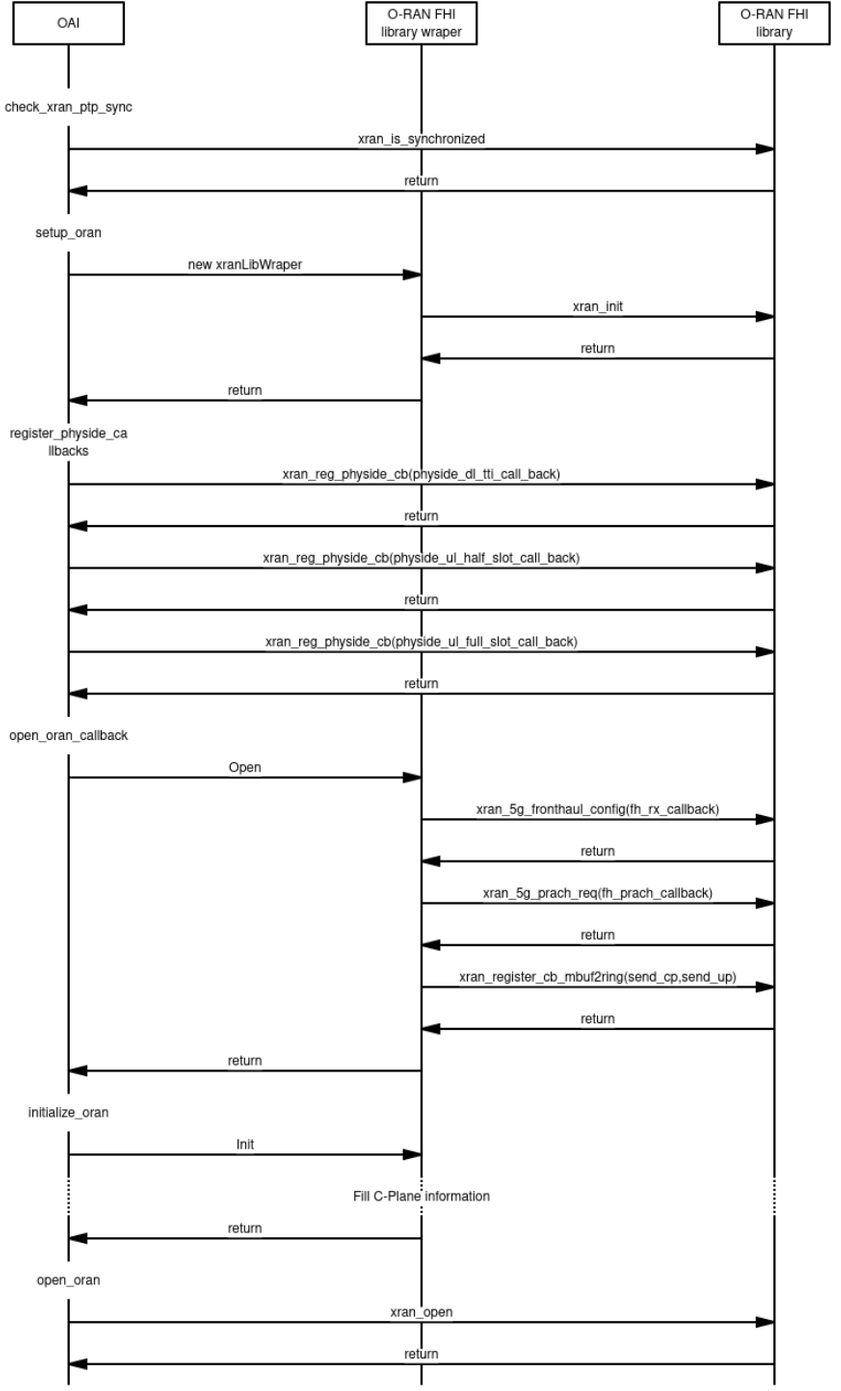


Initialization

Initialization of the library is performed at library loading in the function called *transport_init*.

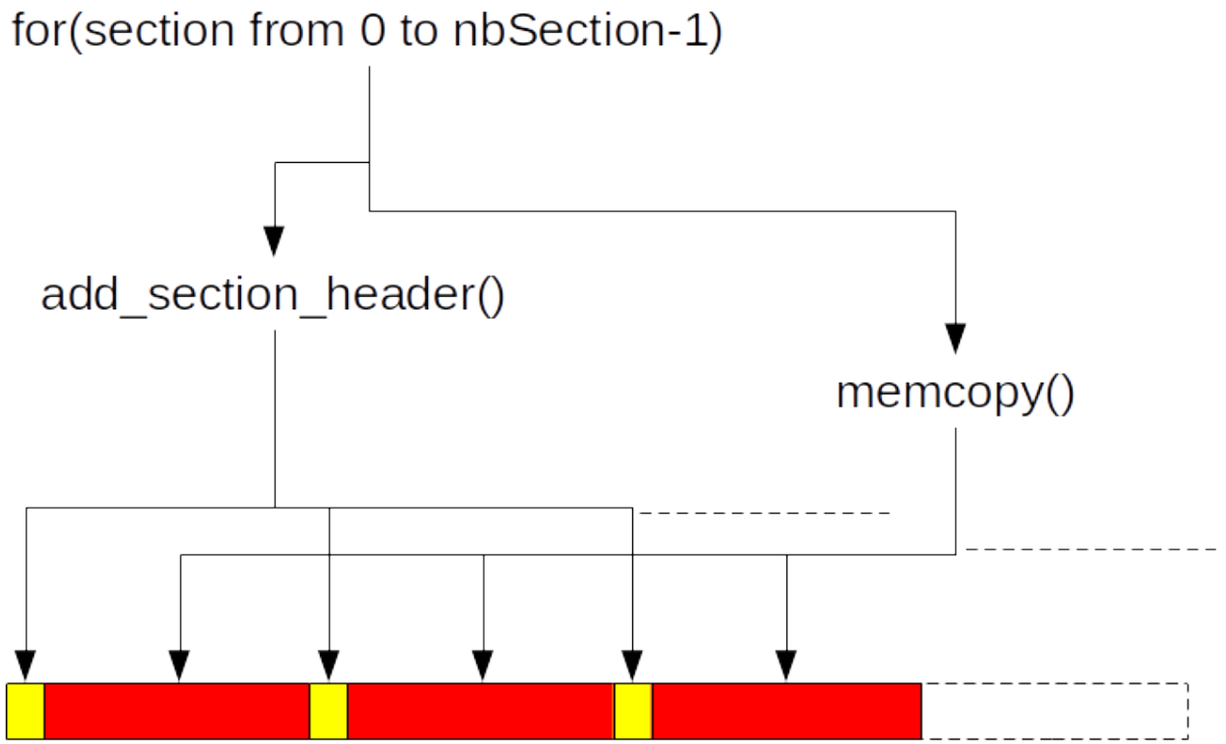
Then OAI calls the function *trx_oran_start* to start the Front Haul





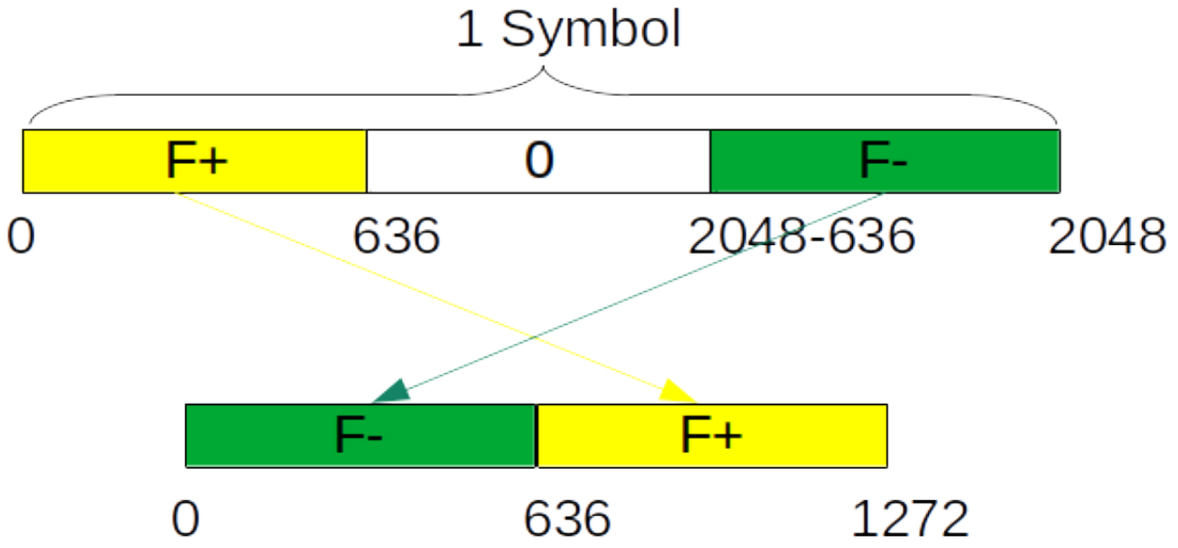
TX implementation

- *oran_fh_if4p5_south_out*
  - For loop for copying one section at a time.
  - Moving the samples from iFFT input shape to transport shape.



TX implementation

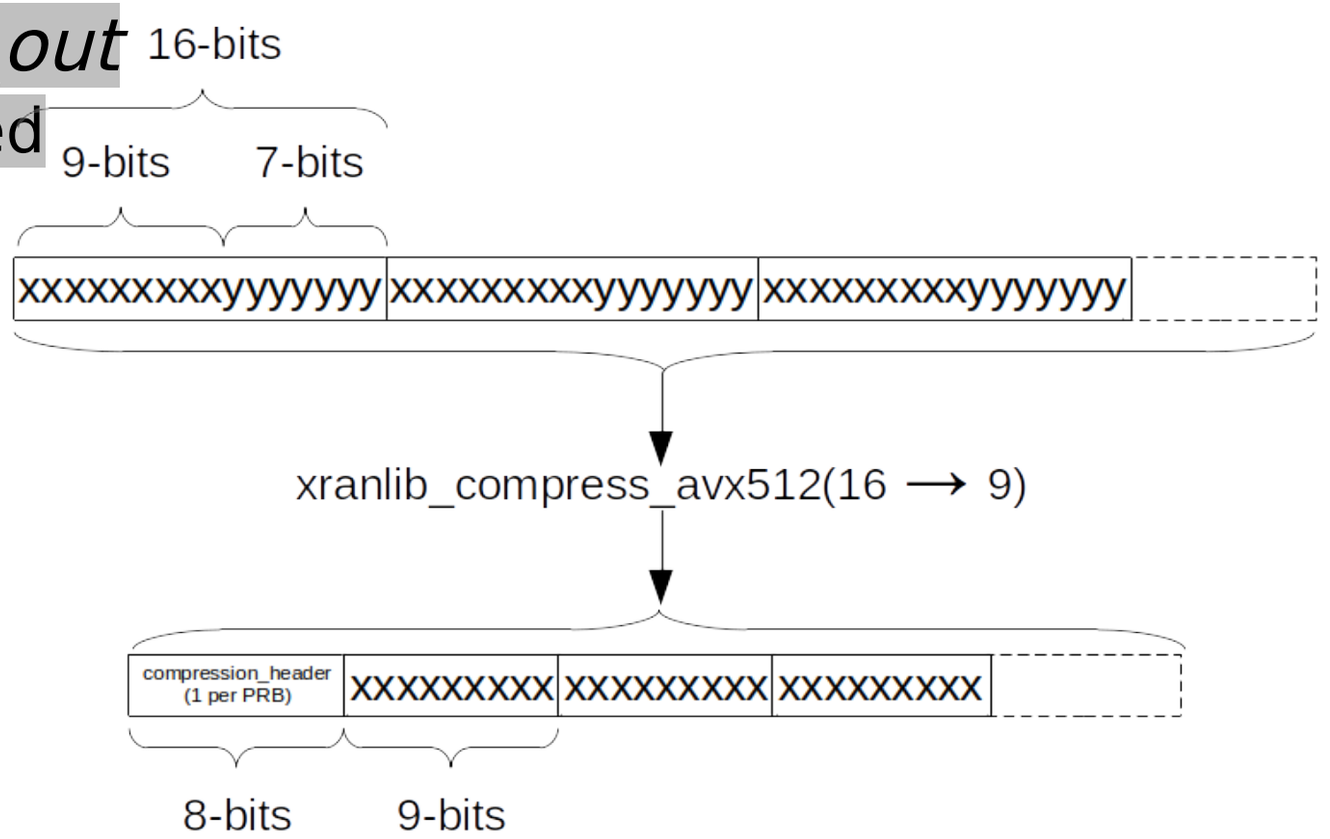
- *oran_fh_if4p5_south_out*
  - For loop for copying one section at a time.
  - Moving the samples from iFFT input shape to transport shape.



Part of the behavior to be verified with a real RU

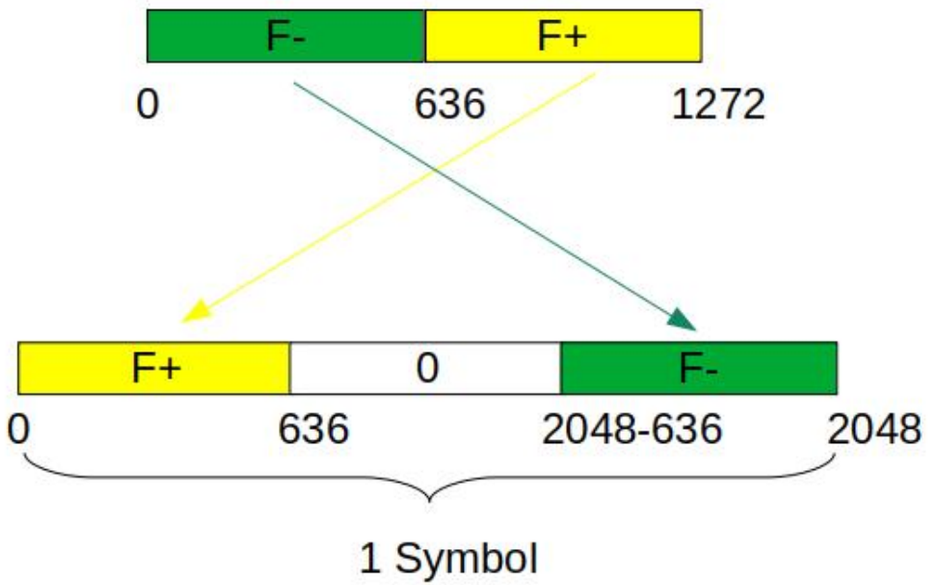
TX implementation: Compression

- *oran_fh_if4p5_south_out*
  - If the option is enabled in configuration, the samples are compressed.



RX Implementation

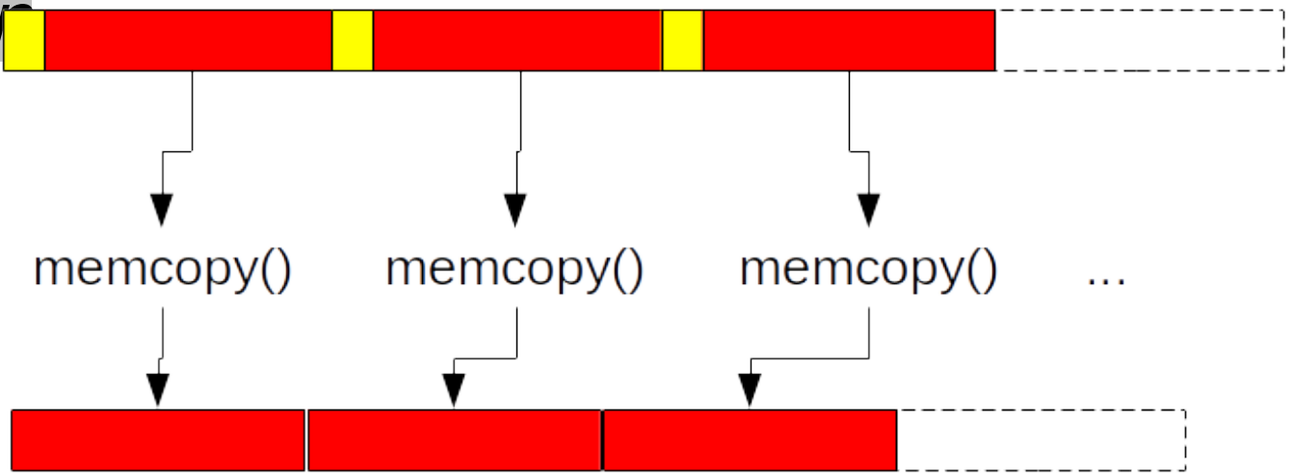
- *oran_fh_if4p5_south*
  - For loop for copying one section at a time.
  - Moving the samples from transport shape to iFFT output shape.



Part of the behavior to be verified with a real RU

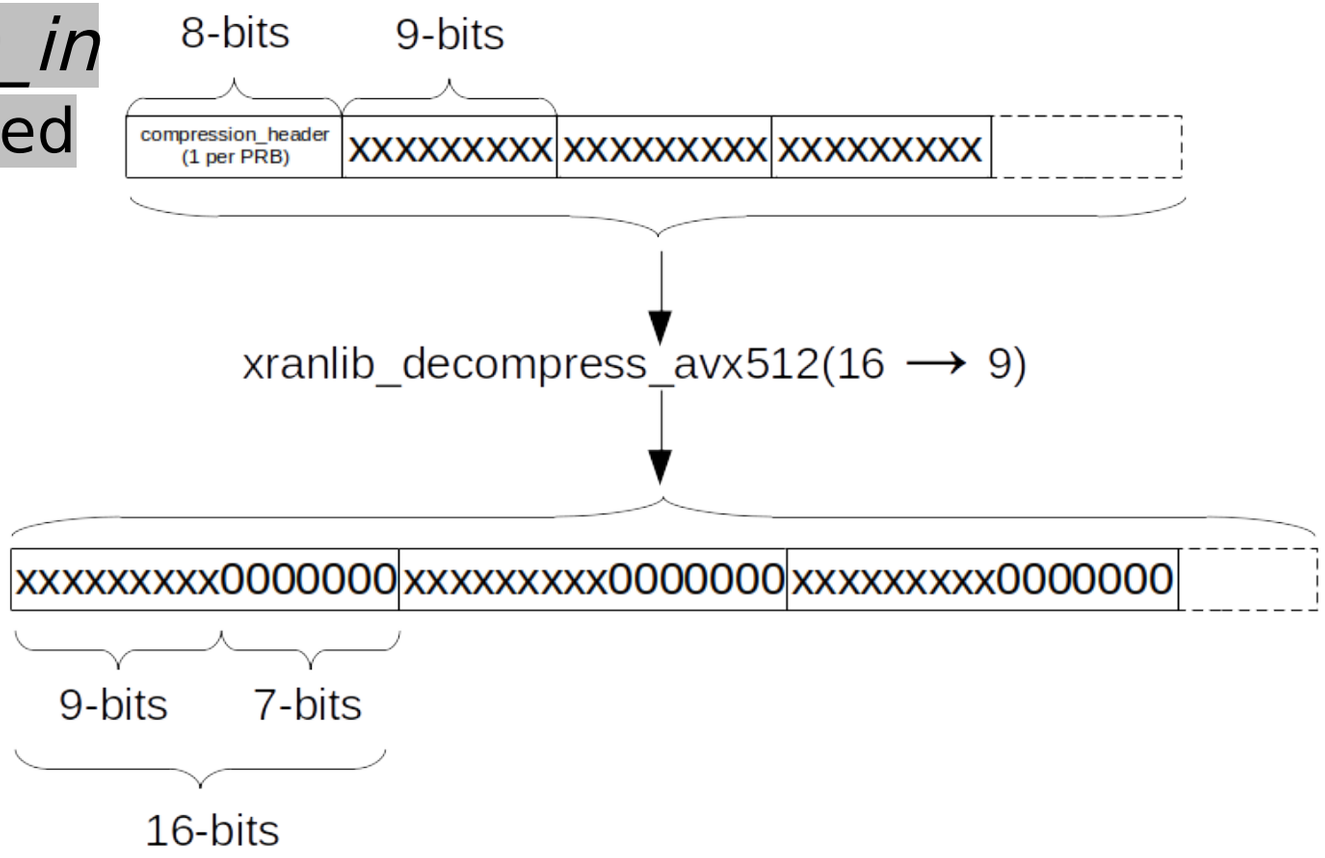
RX Implementation

- *oran_fh_if4p5_south_in*
  - For loop for copying one section at a time.
  - Moving the samples from transport shape to iFFT output shape.



RX Implementation: decompression

- *oran_fh_if4p5_south_in*
  - If the option is enabled in configuration, the samples are uncompressed.



DONE vs TODO: Library integration

- Initialization: Being improved (last version committed is working)
- TX from OAI buffer: Works, timing not aligned.
- RX to OAI buffer: Works, timing not aligned.
- Configuration improvement: Work in progress.
  - Spliting xran_lib_wrap in header and source: mostly done.
  - Removing hardcoded configurations: Work in progress.
  - Unify configuration: TODO (not coming in short term).
- Aligning OAI and library timing
  - Investigate FHI library behavior.
  - Design a way to get reliable library timing in OAI.

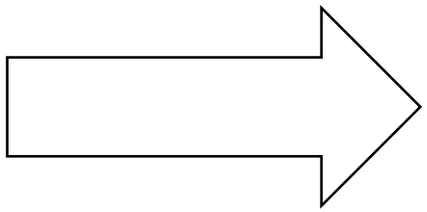
- See: https://hackmd.io/@beurdouc/B1N6LVwct

RX tests

RU side:

0000000 000a 000a 0014 0014 001e 001e
0028 0028
0000010 0032 0032 003c 003c 0046 0046
0050 0050
0000020 005a 005a 0064 0064 006e 006e
0078 0078
0000030 0082 0082 008c 008c 0096 0096
00a0 00a0
0000040 00aa 00aa 00b4 00b4 00be 00be
00c8 00c8
0000050 00d2 00d2 00dc 00dc 00e6 00e6
00f0 00f0
[...]
0000990 17f2 17f2 17fc 17fc 1806 1806
1810 1810
00009a0 181a 181a 1824 1824 182e 182e
1838 1838
00009b0 1842 1842 184c 184c 1856 1856
1860 1860
00009c0 186a 186a 1874 1874 187e 187e
1888 1888
00009d0 1892 1892 189c 189c 18a6 18a6
18b0 18b0
00009e0 18ba 18ba 18c4 18c4 18ce 18ce
18d8 18d8
00009f0 18e2 18e2 18ec 18ec 18f6 18f6
1900 1900
0000a00 190a 190a 1914 1914 191e 191e
1928 1928

9-bit compression, decompression and rearengement

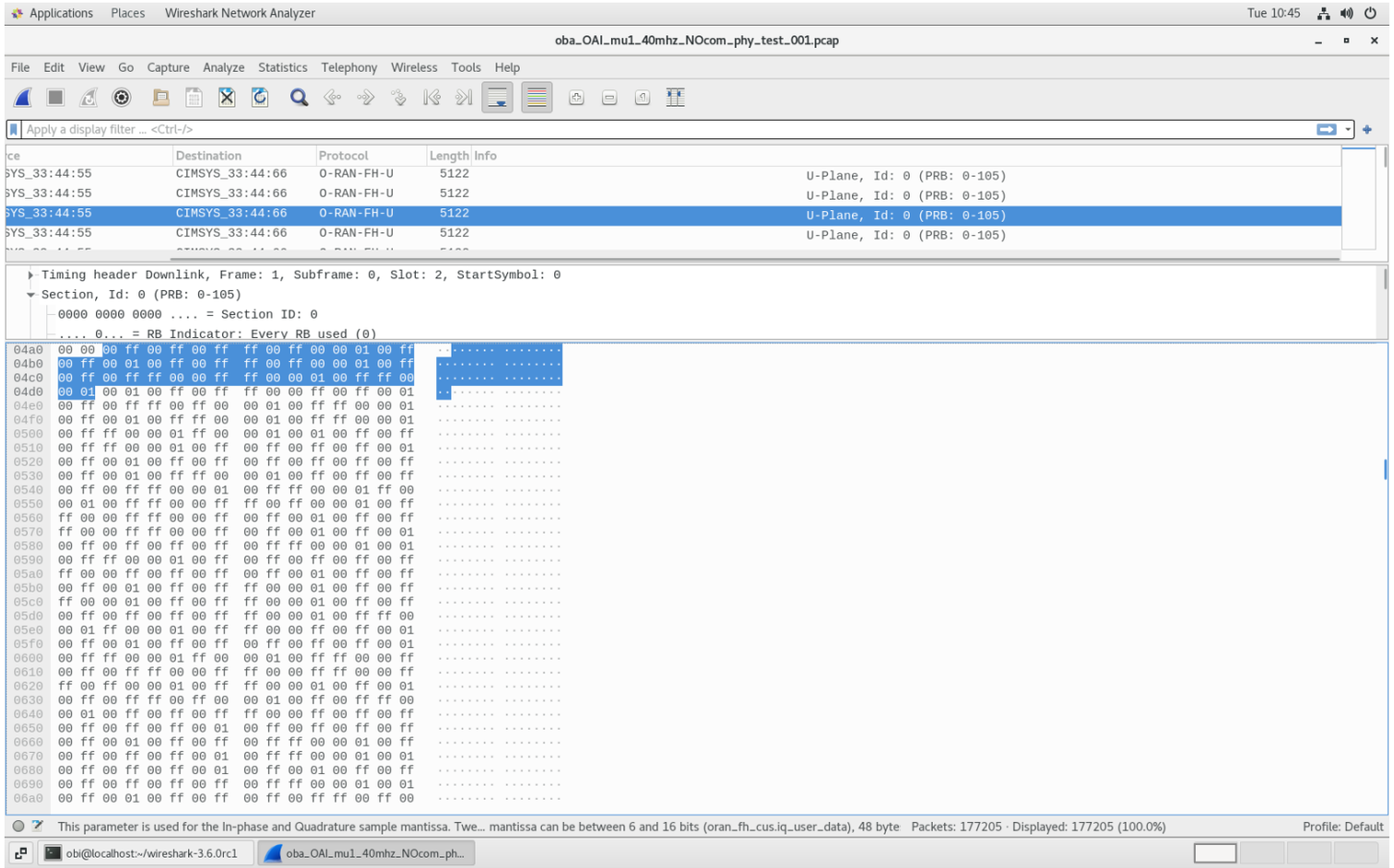


DU side:

e200e200 ec00ec00 f600f600 0
a000a00 14001400 1e001e00 28002800
32003200 3c003c00 46004600 50005000
5a005a00 64006400 6e006e00 78007800
82008200 8c008c00 96009600 a000a000
aa00aa00 b400b400 be00be00 c800c800
[...]
ca00ca00 d400d400 de00de00 e800e800
f200f200 fc00fc00 6000600 10001000
1a001a00 24002400 2e002e00 38003800
42004200 4c004c00 56005600 60006000
6a006a00 74007400 7e007e00 88008800
92009200 9c009c00 a600a600 b000b000
0 0 0 0
[...]
0 0 0 0
a000a00 14001400 1e001e00 28002800
32003200 3c003c00 46004600 50005000
5a005a00 64006400 6e006e00 78007800
82008200 8c008c00 96009600 a000a000
aa00aa00 b400b400 be00be00 c800c800
d200d200 dc00dc00 e600e600 f000f000
[...]
f200f200 fc00fc00 6000600 10001000
1a001a00 24002400 2e002e00 38003800
42004200 4c004c00 56005600 60006000
6a006a00 74007400 7e007e00 88008800
92009200 9c009c00 a600a600 b000b000
ba00ba00 c400c400 ce00ce00 d800d800

TX tests

RU side:



DU side:

oran isolate south_out: frame=1 slot=1 timestamp=645120
1700:ff000100 1701:ffff00 1702:ffff00 1703:ffff00
1704:ff00ff00 1705:ffff00 1706:ff000100
1707:ff00ff00
1708:ff00ff00 1709:ffff00 1710:ffff00
1711:ff000100
1712:ff000100 1713:ffff00 1714:ff000100
1715:ffff00
1716:10000ff 1717:10000ff 1718:ff00ff00
1719:10000ff
1720:ff00ff00 1721:10000ff 1722:10000ff
1723:ff00ff00
1724:ff000100 1725:10000ff 1726:ff00ff00
1727:ffff00
1728:10000ff 1729:ff00ff00 1730:ffff00
1731:ffff00
1732:ff000100 1733:ff00ff00 1734:ff00ff00
1735:10000ff
1736:ffff00 1737:10000ff 1738:10000ff
1739:ffff00
1740:ffff00 1741:ff000100 1742:ffff00
1743:ff000100
1744:ffff00 1745:ffff00 1746:ffff00 1747:ff000100
1748:10000ff 1749:ff00ff00 1750:10000ff
1751:ff00ff00
1752:10000ff 1753:ff00ff00 1754:ffff00
1755:ff00ff00
1756:ff00ff00 1757:ff00ff00 1758:ffff00
1759:ff000100
1760:10000ff 1761:ff00ff00 1762:ff00ff00
1763:ff000100
1764:ff000100 1765:ff00ff00 1766:ff00ff00
1767:ffff00
1768:ff00ff00 1769:ffff00 1770:10000ff 1771:ffff00
1772:10000ff 1773:ffff00 1774:ffff00 1775:ff00ff00
1776:ff00ff00 1777:ffff00 1778:ffff00 1779:ff00ff00
1780:ff00ff00 1781:ffff00 1782:ffff00
1783:ff000100
1784:ff00ff00 1785:ff00ff00 1786:ffff00

DONE vs TODO

- Understanding the library is mostly done
- The test application `wrapper.cpp` is mostly finished.
- Integration is in progress
  - Initialization: Being improved (last version committed is working)
  - First try to reproduce sample app behaviour: done
  - TX from OAI buffer: Works, timing not aligned.
  - RX to OAI buffer: Works, timing not aligned.
  - Configuration improvement: Work in progress

- See: https://hackmd.io/@beurdouc/B1N6LVwct

Links

- Summary of O-RAN FrontHaul Library project
  - https://hackmd.io/@beurdouc/Hk5CVHM_Y
- Issues List
  - https://hackmd.io/@beurdouc/By5DR6g9F
- Results
  - https://hackmd.io/@beurdouc/Ski00eu0Y

Outdated slides from Here

Running the Test Application

- Receiving Data: OK
  - By printing the non null data, we manage to identify an eCPRI packet
  - We still have to figure out how exactly this data should be interpreted and passed to upper laye (comes after the following issue).
- Sending Data: NON OK
  - We managed to show that the machine running `wrapper` was not outputing anything on its ethernet port.
  - In the same condition, the provided `sample-app` works.

Investigating the Issue

- Possible causes we thought about
  - Bad Configuration:
    - `wrapper` configuration is done with a function defined in `xran_lib_wrap.hpp` that parses a json file: `conf.json`.
    - We already found parsing errors in this function, there may be others.
    - `sample-app` configuration is done using another function defined in the library which parses another file.
  - Bad use of the API:
    - We may not be using the API as it is supposed to be used.

Integration in OAI

- Basic Initialization: OK
  - We took as a starting point an existing library wrapper for another FrontHaul implementation.
  - We kept only the initialization function and we made the interface function blank.
  - The ORAN FHI Library initialization was done in the starting function of the interface.
- Further function integration: In Progress
  - It requires first to master the API, which is done by trying to run the wrapper test application.

Major Issue: Lack of Guidance

- A few Resources about the Library are available:
  - The online documentation: https://docs.o-ran-sc.org/projects/o-ran-sc-o-du-phy/en/latest/Introduction_fh.html contains lots of information about the architecture of the library but no precise information on how to use it.
  - API documentation can be generated using Doxygen but it does not contain any explanation, only interface descriptions.
  - The provided applications (e.g. sample-app) seems to work but their code are difficult to analyze. A deep analysis show that they are now obsolete due to changes in the API over time.
- In Conclusion:

  We are lacking Reliable Guidance about the way to use the API!