FLEXRAN 4G LTE & 5G NR: TESTMAC

NPG Wireless Access Division

Minjun Li

Oct 2018

Legal Disclaimer

General Disclaimer:

Technology Disclaimer:

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at [intel.com].

Performance Disclaimers (include only the relevant ones):

Cost reduction scenarios described are intended as examples of how a given Intel- based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

Results have been estimated or simulated using internal Intel analysis or architecture simulation or modeling, and provided to you for informational purposes. Any differences in your system hardware, software or configuration may affect your actual performance.

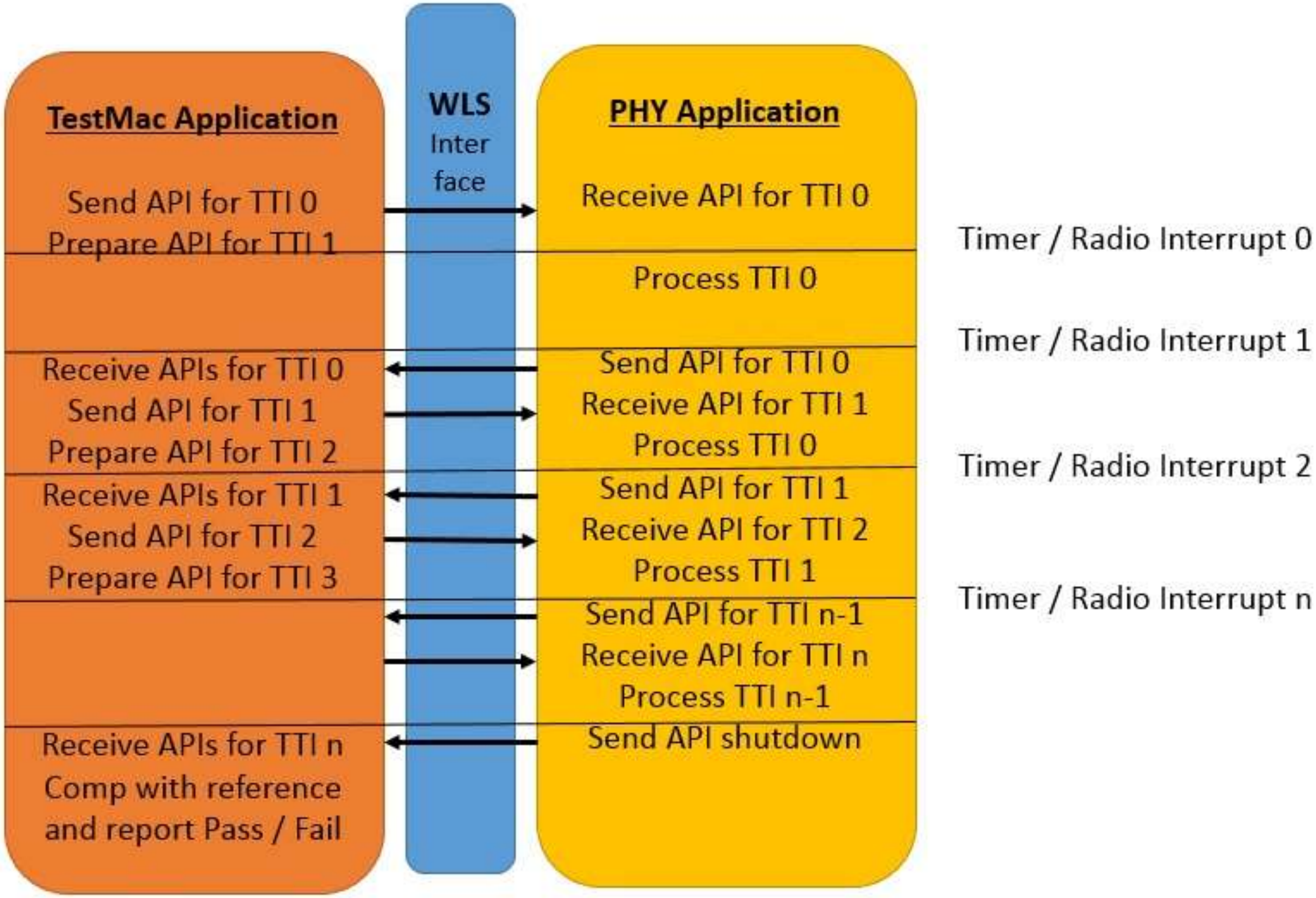
Agenda

- Test MAC for 4G and 5G

Test MAC

- Test MAC is an application used to test L1 functionality
- Prepares and delivers PHY APIs in the same manner as real L2
- Use wls_libs calls so that from the point of view of the L1, it does not know the difference between testmac and real L2 mac.
- Setup custom configurations for downlink only, uplink only or full duplex test cases.
  - All downlink and uplink channels and signals can be tested.
- Can run in real time or non-real time mode.
  - Non-real time mode can be used for debugging issues in L1 by using file prints and prints to screen.
- Framework or bbu pooling will dispatch tasks in the same manner when running testmac tests and real L2 stack.

Test MAC



TestMac Application
Send API for TTI 0
Prepare API for TTI 1
Receive APIs for TTI 0
Send API for TTI 1
Prepare API for TTI 2
Receive APIs for TTI 1
Send API for TTI 2
Prepare API for TTI 3
Receive APIs for TTI n
Comp with reference
and report Pass / Fail
WLS
Inter
face
PHY Application
Receive API for TTI 0
Process TTI 0
Send API for TTI 0
Receive API for TTI 1
Process TTI 0
Send API for TTI 1
Receive API for TTI 2
Process TTI 1
Send API for TTI n-1
Receive API for TTI n
Process TTI n-1
Send API shutdown
Timer / Radio Interrupt 0
Timer / Radio Interrupt 1
Timer / Radio Interrupt 2
Timer / Radio Interrupt n

Test MAC

- Source code for test mac located in:
  - source/test/testmac
- Application can only be built in a Linux environment using icc 18.0.1
- To compile test mac project:
  - Launch ./build.sh located in:
    - build/nr5g/gnb/testmac for 5g NR
    - build/lte/testmac for 4G LTE
    - ./build.sh xclean ->To clean all object
- The executable (testmac) will be in:
  - /bin/lte/testmac folder for 4g and /bin/nr5g/gnb/testmac folder for 5g.
- Compile l1app with the following script:
  - Launch ./build.sh located in: build/lte/l1app for 4g
  - Launch ./build_phy.sh located in: build/nr5g/gnb/l1app for 5g
- Must add export command:
  - export DIR_WIRELESS_TEST_4G=path to the 4g test repo
  - export DIR_WIRELESS_TEST_5G=path to the 5g test repo

Test MAC for 4G

- After compiling l1app, go to the /bin/lte/l1 directory and launch:
  - ./l1.sh –e for timer mode or ./l1.sh for radio mode
  - Phycfg.xml will be used to configure L1 for radio mode.
  - Phycfg_timer.xml will be used to configure L1 for timer mode.
- Launch a second ssh session and go the /bin/lte/testmac directory and launch:
  - ./l2.sh
    - Optional parameter in command line:
    - --testfile=test_to_run.cfg where the .cfg file will run a test specified in the .cfg file.   Used to help automate testing.
    - Sampe Test_to_run.cfg:
      - TEST_DL, 2, 1, dl/2/dl_testconfig_tst2.cfg
      - '2' – tests number
      - '1' – number of carriers to run
      - Path to the test config file.

Test MAC for 5G

- After compiling l1app, go to the /bin/nr5g/gnb/l1 directory and launch:
  - ./l1.sh or ./l1.sh –r for radio mode
  - The –r is there is to signify radio mode. Phycfg.xml will be used to configure L1.
  - Without any arguments, phycfg_timer.xml will be used to configure L1.
- Launch a second ssh session and go the /bin/nr5g/gnb/testmac directory and launch:
  - ./l2.sh
    - Optional parameter in command line:
    - --testfile=test_to_run.cfg where the .cfg file will run a test specified in the .cfg file. Used to help automate testing.
    - Sampe Test_to_run.cfg:
      - TEST_DL, 2, 1, dl/2/dl_testconfig_tst2.cfg
      - ‘2’ – tests number
      - ‘1’ – number of carriers to run
      - Path to the test config file.

Test MAC

- In the testmac SSH session, a TESTMAC prompt should show up.
- Commands:
    - phystart mode tti_interval num_tti
        - mode:
            - 0 = radio mode. FH FPGA enabled. TTI will be sent by FPGA.
            - 1 = timer mode. FH FPGA disabled.   TTI will be sent by timer thread.
        - tti_interval is the TTI duration. If set to 1, it means that it will run in real time with the TTI being calculated by numerology.   Can set to any positive integer to increase TTI interval for debugging purposes.
        - num_tti is the number of TTIs to run.   If set to 0, it will run the test as long as the test configuration file will tell it to run.   Each pre-defined test case comes with a default number of slots to run.
        - Default settings are phystart 1 10 0.
    - run testtype testnum
        - Testtype, 0 for DL only, 1 for UL only, 2 for FD (full duplex)
        - Testnum is the pre-defined test scenarios included in release.

Test MAC

- Example of how to run Downlink Test Case 3 for regression Unit testing:
  - SSH window in the /bin/nr5g/gnb/l1 folder for 5G or /bin/lte/l1 for 4G:
    - ./l1.sh
  - SSH window in the /bin/nr5g/gnb/testmac folder for 5G or /bin/lte/testmac for 4G
    - ./l2.sh
  - In the testmac window:
    - run 0 3
  - This will run with default settings of:
    - phystart 1 10 0
      - 1 -> timer mode
      - 10 -> 10 times normal TTI length
      - 0 -> number of slots taken from test config file
  - The result of the test will show up in the testmac window:
    - 3 | Result: PASS | PDSCH: P | PUSCH: - | RXBITS: - | PUCCH: - | MUXSCH: - | SNR: - | TA: - | RACH: - | SRS: - |
    - All Tests Completed

Test MAC

- Example of how to run Downlink Test Case 3 for checking cycles:
  - When using testmac to check for cycles, the number of slots to run must be increased.
  - Repeat steps by launching ./l1.sh and ./l2.sh in respective windows.
  - To do so:
    - phystart 1 1 5000
      - 1 -> timer mode
      - 1 -> use actual TTI length depending on numerology
      - 5000 -> number of slots to run
  - Test will now run for 5000 slots.   The configuration will repeat the number of slots configured in config files.
  - There will not be a pass/fail criteria when running in this mode since the reference IQ samples stored is predetermined by the number of slots in config file.

Test MAC

- Example of how to run Downlink Test Case 3 with real RRH:
  - TestMAC test cases can be used to send signal with real RRH.
  - Repeat steps by launching
    - ./l1.sh –r   (notice the –r as input argument, for 5G)
    - ./l1.sh   (no argument for 4G)
    - ./l2.sh
    - To do so:
    - phystart 0 0 0
      - First 0 -> radio mode
      - When in radio mode, it will ignore the other two parameters
    - run 0 3
  - Prints will come to the screen after a short while.
  - Connect UXA to see constellation.

Test MAC

- Similar steps to run Uplink tests and full duplex tests.
  - To run Uplink tests, simply replace the ‘0’ with ‘1’.
  - For example, to run Uplink Test Case 2:
    - run 1 2
  - To run Full Duplex tests, replace the ‘0’ with ‘2’.
  - For example, to run Full Duplex Test Case 4:
    - run 2 104
    - Notice that the test case 4 became ‘104’ for test case number.
    - For Full Duplex test cases, the first digit represents number of cells.
    - So to run Full Duplex Test Case 4 with 3 cells:
      - run 2 304
- All test scenarios described for Downlink testing is valid for Uplink and Full Duplex testing.

intel®

experience
what's inside™