

Intel testing coverage not currently running at UNH lab:

TLDR: For DTS, we have basically the same hardware and could likely add the functional tests and functional-smoke tests Intel Lab is using to our testruns. For compile testing, we mostly use the same distros, but a few gaps there can be easily bridged. The most interesting gap in compile testing is we almost exclusively compile with GCC and static library linking, whereas Intel Lab has more diversity like more extensive use of Clang than us, more extensive use of shared libraries vs static, and some use of debug build type.

From referencing the Intel test coverage and the UNH Lab test coverage, the below testing is being run by the Intel lab and not being run by the UNH Community Lab. If redundancy is desired from the community, UNH may be able to run these at the Community Lab. Hardware compatibility and test capacity concerns may have to be evaluated on a case by case basis, but likely most of the coverage below could be replicated at the Community Lab.

DTS. Bullets are testsuites. Sub-bullets are testcases

NIC : Ethernet Controller XL710 for 40GbE QSFP+ (UNH also runs this NIC for DTS)

- **Checksum_offload:**
https://git.dpdk.org/tools/dts/tree/test_plans/checksum_offload_test_plan.rst
 - test_checksum_offload_with_vlan
 - test_do_not_insert_checksum_on_the_transmit_packet
 - test_hardware_checksum_check_ip_rx
 - test_hardware_checksum_check_ip_tx
 - test_hardware_checksum_check_l4_rx
 - test_hardware_checksum_check_l4_tx
 - test_insert_checksum_on_the_transmit_packet
 - Test_rx_checksum_valid_flags
- **Dual_vlan:** https://git.dpdk.org/tools/dts/tree/test_plans/dual_vlan_test_plan.rst
 - test_vlan_filter_config
 - Test_vlan_filter_table
 - Test_vlan_insert_config
 - Test_vlan_random_test
 - test_vlan_strip_config
 - test_vlan_synthetic_test
 - Test_vlan_tpid_config
- **Jumbo Frames:**
https://git.dpdk.org/tools/dts/tree/test_plans/jumboframes_test_plan.rst
 - test_jumboframes_bigger_jumbo
 - test_jumboframes_jumbo_jumbo
 - test_jumboframes_jumbo_nojumbo
 - test_jumboframes_normal_jumbo
 - Test_jumboframes_normal_nojumbo
- **Rxtx_offload:**
https://git.dpdk.org/tools/dts/tree/test_plans/rxtx_offload_test_plan.rst
 - test_rxoffload_port_all
 - test_rxoffload_port_cmdline

- test_txoffload_port
- test_txoffload_port_all
- test_txoffload_port_checksum
- test_txoffload_port_cmdline
- test_txoffload_port_multi_segs
- Test_txoffload_queue

NIC : Ethernet Controller E810-C for SFP. UNH has an E810-C from the 2023 hardware refresh. UNH has the QSFP model, not SFP, but I'm guessing the cabling is not very significant though.

- **Asan_smoke:** https://git.dpdk.org/tools/dts/tree/test_plans/asan_smoke_test_plan.rst
 - test_rtx_with_ASAN_enable
- **Pf_smoke:** https://git.dpdk.org/tools/dts/tree/test_plans/pf_smoke_test_plan.rst
 - Test_pf_jumbo_frames
 - test_pf_rss
 - Test_pf_tx_rx_queue
- **Vf_smoke:** https://git.dpdk.org/tools/dts/tree/test_plans/vf_smoke_test_plan.rst
 - test_vf_rss
 - Test_vf_tx_rx_queue
 - test_vf_jumbo_frames

Compile testing: We mostly use the same distros for our compile testing. The main difference is that we almost exclusively compile with GCC, and almost exclusively compile with static library linking instead of shared. They also include some more variety in build_type options used (debug in some cases).

OS/distros + compiler combos:

- *OpenAnolis8.8-64*
 - GCC-static
- *FreeBSD13-64*
 - Clang-static
 - GCC-shared
 - GCC-debug
- Rhel-9 (we are already adding this distro for gcc-static soon independent of any requests)
 - GCC-static
 - Clang-static
 - GCC-shared
 - GCC-debug
- *CBL-Mariner2.0-64*
 - GCC-static
- Ubuntu 22.04 (32-bit)
 - GCC-static
- Ubuntu 22.04 (64-bit)
 - Clang-static
 - Document?
 - GCC-16byte
- Ubuntu 23.04
 - GCC-static
- Rhel-9-rt
 - GCC-static

- Ubuntu 22.04-rt
 - GCC-static

MISC:

1. I dont know what “document” means for the Ubuntu 22.04 compile job.
2. I think mentions of “rt” above mean “real time.” So, real time linux for Ubuntu and RHEL-9