DPDK Release Notes

Release 2.1.0

August 17, 2015

1 Description of Release **2**

2 DPDK Release 2.1 **3**
2.1 New Features . . . 3
2.2 Resolved Issues . . . 9
2.3 Known Issues . . . 16
2.4 API Changes . . . 16
2.5 ABI Changes . . . 16

3 DPDK Release 2.0 **17**
3.1 New Features . . . 17

4 DPDK Release 1.8 **19**
4.1 New Features . . . 19

5 Supported Operating Systems **20**

6 Known Issues and Limitations in Legacy Releases **21**
6.1 Unit Test for Link Bonding may fail at test_tlb_tx_burst() . . . 21
6.2 Pause Frame Forwarding does not work properly on igb . . . 21
6.3 In packets provided by the PMD, some flags are missing . . . 21
6.4 The rte_malloc library is not fully implemented . . . 22
6.5 HPET reading is slow . . . 22
6.6 HPET timers do not work on the Osage customer reference platform . . . 22
6.7 Not all variants of supported NIC types have been used in testing . . . 23
6.8 Multi-process sample app requires exact memory mapping . . . 23
6.9 Packets are not sent by the 1 GbE/10 GbE SR-IOV driver when the source MAC is not the MAC assigned to the VF NIC . . . 24
6.10 SR-IOV drivers do not fully implement the rte_ethdev API . . . 24
6.11 PMD does not work with –no-huge EAL command line parameter . . . 25
6.12 Some hardware off-load functions are not supported by the VF Driver . . . 25
6.13 Kernel crash on IGB port unbinding . . . 25
6.14 Twinpond and Ironpond NICs do not report link status correctly . . . 26
6.15 Discrepancies between statistics reported by different NICs . . . 26
6.16 Error reported opening files on DPDK initialization . . . 26
6.17 Intel® QuickAssist Technology sample application does not work on a 32-bit OS on Shumway . . . 26
6.18 IEEE1588 support possibly not working with an Intel® Ethernet Controller I210 NIC . . . 27

6.19 Differences in how different Intel NICs handle maximum packet length for jumbo frame . . . 27
6.20 Binding PCI devices to igb_uio fails on Linux kernel 3.9 when more than one device is used . . . 27
6.21 GCC might generate Intel® AVX instructions for processors without Intel® AVX support . . . 28
6.22 Ethertype filter could receive other packets (non-assigned) in Niantic . . . 28
6.23 Cannot set link speed on Intel® 40G Ethernet controller . . . 28
6.24 Stopping the port does not down the link on Intel® 40G Ethernet controller . . . 29
6.25 Devices bound to igb_uio with VT-d enabled do not work on Linux kernel 3.15-3.17 29
6.26 VM power manager may not work on systems with more than 64 cores . . . 29

7 ABI and API Deprecation **30**
7.1 Deprecation Notices . . . 30

August 17, 2015

Contents

CHAPTER

ONE

DESCRIPTION OF RELEASE

This document contains the release notes for Data Plane Development Kit (DPDK) release version 2.1.0 and previous releases.

It lists new features, fixed bugs, API and ABI changes and known issues.

For instructions on compiling and running the release, see the DPDK Getting Started Guide.

CHAPTER

TWO

DPDK RELEASE 2.1

2.1 New Features

- **Enabled cloning of indirect mbufs.**

  This feature removes a limitation of `rte_pktmbuf_attach()` which generated the warning: “mbuf we’re attaching to must be direct”.

  Now, when attaching to an indirect mbuf it is possible to:

  - Copy all relevant fields (address, length, offload, ...) as before.
  - Get the pointer to the mbuf that embeds the data buffer (direct mbuf), and increase the reference counter.

  When detaching the mbuf, we can now retrieve this direct mbuf as the pointer is determined from the buffer address.

- **Extended packet type support.**

  In previous releases mbuf packet types were indicated by 6 bits in the `ol_flags`. This was not enough for some supported NICs. For example i40e hardware can recognize more than 150 packet types. Not being able to identify these additional packet types limits access to hardware offload capabilities

  So an extended “unified” packet type was added to support all possible PMDs. The 16 bit packet_type in the mbuf structure was changed to 32 bits and used for this purpose.

  To avoid breaking ABI compatibility, the code changes for this feature are enclosed in a `RTE_NEXT_ABI` ifdef. This is enabled by default but can be turned off for ABI compatibility with DPDK R2.0.

- **Reworked memzone to be allocated by malloc and also support freeing.**

  In the memory hierarchy, memsegs are groups of physically contiguous hugepages, memzones are slices of memsegs, and malloc slices memzones into smaller memory chunks.

  This feature modifies `malloc()` so it partitions memsegs instead of memzones. Now memzones allocate their memory from the malloc heap.

  Backward compatibility with API and ABI are maintained.

  This allow memzones, and any other structure based on memzones, for example mempools, to be freed. Currently only the API from freeing memzones is supported.

- **Interrupt mode PMD.**

  This feature introduces a low-latency one-shot RX interrupt into DPDK. It also adds a polling and interrupt mode switch control example.

  DPDK userspace interrupt notification and handling mechanism is based on UIO/VFIO with the following limitations:

  - Per queue RX interrupt events are only allowed in VFIO which supports multiple MSI-X vectors.
  - In UIO, the RX interrupt shares the same vector with other interrupts. When the RX interrupt and LSC interrupt are both enabled, only the former is available.
  - RX interrupt is only implemented for the linuxapp target.
  - The feature is only currently enabled for tow PMDs: ixgbe and igb.

- **Packet Framework enhancements.**

  Several enhancements were made to the Packet Framework:

  - A new configuration file syntax has been introduced for IP pipeline applications. Parsing of the configuration file is changed.
  - Implementation of the IP pipeline application is modified to make it more structured and user friendly.
  - Implementation of the command line interface (CLI) for each pipeline type has been moved to the separate compilation unit. Syntax of pipeline CLI commands has been changed.
  - Initialization of IP pipeline is modified to match the new parameters structure.
  - New implementation of pass-through pipeline, firewall pipeline, routing pipeline, and flow classification has been added.
  - Master pipeline with CLI interface has been added.
  - Added extended documentation of the IP Pipeline.

- **Added API for IEEE1588 timestamping.**

  This feature adds an ethdev API to enable, disable and read IEEE1588/802.1AS PTP timestamps from devices that support it. The following functions were added:

  - `rte_eth_timesync_enable()`
  - `rte_eth_timesync_disable()`
  - `rte_eth_timesync_read_rx_timestamp()`
  - `rte_eth_timesync_read_tx_timestamp()`

  The “ieee1588” forwarding mode in testpmd was also refactored to demonstrate the new API.

- **Added multicast address filtering.**

  Added multicast address filtering via a new ethdev function `set_mc_addr_list()`.

  This overcomes a limitation in previous releases where the receipt of multicast packets on a given port could only be enabled by invoking the `rte_eth_allmulticast_enable()` function. This method did not work for VFs

in SR-IOV architectures when the host PF driver does not allow these operation on VFs. In such cases, joined multicast addresses had to be added individually to the set of multicast addresses that are filtered by the [VF] port.

- **Added Flow Director extensions.**

  Several Flow Director extensions were added such as:

  - Support for RSS and Flow Director hashes in vector RX.
  - Added Flow Director for L2 payload.

- **Added RSS hash key size query per port.**

  This feature supports querying the RSS hash key size of each port. A new field `hash_key_size` has been added in the `rte_eth_dev_info` struct for storing hash key size in bytes.

- **Added userspace ethtool support.**

  Added userspace ethtool support to provide a familiar interface for applications that manage devices via kernel-space `ethtool_op` and `net_device_op`.

  The initial implementation focuses on operations that can be implemented through existing `netdev` APIs. More operations will be supported in later releases.

- **Updated the ixgbe base driver.**

  The ixgbe base driver was updated with several changes including the following:

  - Added a new 82599 device id.
  - Added new X550 PHY ids.
  - Added SFP+ dual-speed support.
  - Added wait helper for X550 IOSF accesses.
  - Added X550em features.
  - Added X557 PHY LEDs support.
  - Commands for flow director.
  - Issue firmware command when resetting X550em.

  See the git log for full details of the ixgbe/base changes.

- **Added additional hotplug support.**

  Port hotplug support was added to the following PMDs:

  - e1000/igb.
  - ixgbe.
  - i40e.
  - fm10k.
  - ring.
  - bonding.
  - virtio.

Port hotplug support was added to BSD.

- **Added ixgbe LRO support.**

  Added LRO support for x540 and 82599 devices.

- **Added extended statistics for ixgbe.**

  Implemented `xstats_get()` and `xstats_reset()` in dev_ops for ixgbe to expose detailed error statistics to DPDK applications.

  These will be implemented for other PMDs in later releases.

- **Added proc_info application.**

  Created a new `proc_info` application, by refactoring the existing `dump_cfg` application, to demonstrate the usage of retrieving statistics, and the new extended statistics (see above), for DPDK interfaces.

- **Updated the i40e base driver.**

  The i40e base driver was updated with several changes including the following:

  - Support for building both PF and VF driver together.
  - Support for CEE DCBX on recent firmware versions.
  - Replacement of `i40e_debug_read_register()`.
  - Rework of `i40e_hmc_get_object_va`.
  - Update of shadow RAM read/write functions.
  - Enhancement of polling NVM semaphore.
  - Enhancements on adminq init and sending asq command.
  - Update of get/set LED functions.
  - Addition of AOC phy types to case statement in get_media_type.
  - Support for iSCSI capability.
  - Setting of FLAG_RD when sending driver version to FW.

  See the git log for full details of the i40e/base changes.

- **Added support for port mirroring in i40e.**

  Enabled mirror functionality in the i40e driver.

- **Added support for i40e double VLAN, QinQ, stripping and insertion.**

  Added support to the i40e driver for offloading double VLAN (QinQ) tags to the mbuf header, and inserting double vlan tags by hardware to the packets to be transmitted. Added a new field `vlan_tci_outer` in the `rte_mbuf` struct, and new flags in `ol_flags` to support this feature.

- **Added fm10k promiscuous mode support.**

  Added support for promiscuous/allmulticast enable and disable in the fm10k PF function. VF is not supported yet.

- **Added fm10k jumbo frame support.**

  Added support for jumbo frame less than 15K in both VF and PF functions in the fm10k pmd.

- **Added fm10k mac vlan filtering support.**

  Added support for the fm10k MAC filter, only available in PF. Updated the VLAN filter to add/delete one static entry in the MAC table for each combination of VLAN and MAC address.

- **Added support for the Broadcom bnx2x driver.**

  Added support for the Broadcom NetXtreme II bnx2x driver. It is supported only on Linux 64-bit and disabled by default.

- **Added support for the Chelsio CXGBE driver.**

  Added support for the CXGBE Poll Mode Driver for the Chelsio Terminator 5 series of 10G/40G adapters.

- **Enhanced support for Mellanox ConnectX-3 driver (mlx4).**
  - Support Mellanox OFED 3.0.
  - Improved performance for both RX and TX operations.
  - Better link status information.
  - Outer L3/L4 checksum offload support.
  - Inner L3/L4 checksum offload support for VXLAN.

- **Enabled VMXNET3 vlan filtering.**

  Added support for the VLAN filter functionality of the VMXNET3 interface.

- **Added support for vhost live migration.**

  Added support to allow live migration of vhost. Without this feature, qemu will report the following error: “migrate: Migration disabled: vhost lacks VHOST_F_LOG_ALL feature”.

- **Added support for pcap jumbo frames.**

  Extended the PCAP PMD to support jumbo frames for RX and TX.

- **Added support for the TILE-Gx architecture.**

  Added support for the EZchip TILE-Gx family of SoCs.

- **Added hardware memory transactions/lock elision for x86.**

  Added the use of hardware memory transactions (HTM) on fast-path for rwlock and spinlock (a.k.a. lock elision). The methods are implemented for x86 using Restricted Transactional Memory instructions (Intel(r) Transactional Synchronization Extensions). The implementation fall-backs to the normal rwlock if HTM is not available or memory transactions fail. This is not a replacement for all rwlock usages since not all critical sections protected by locks are friendly to HTM. For example, an attempt to perform a HW I/O operation inside a hardware memory transaction always aborts the transaction since the CPU is not able to roll-back should the transaction fail. Therefore, hardware transactional locks are not advised to be used around `rte_eth_rx_burst()` and `rte_eth_tx_burst()` calls.

- **Updated Jenkins Hash function**

  Updated the version of the Jenkins Hash (jhash) function used in DPDK from the 1996 version to the 2006 version. This gives up to 35% better performance, compared to the original one.

  Note, the hashes generated by the updated version differ from the hashes generated by the previous version.

- **Added software implementation of the Toeplitz RSS hash**

  Added a software implementation of the Toeplitz hash function used by RSS. It can be used either for packet distribution on a single queue NIC or for simulating RSS computation on a specific NIC (for example after GRE header de-encapsulation).

- **Replaced the existing hash library with a Cuckoo hash implementation.**

  Replaced the existing hash library with another approach, using the Cuckoo Hash method to resolve collisions (open addressing). This method pushes items from a full bucket when a new entry must be added to it, storing the evicted entry in an alternative location, using a secondary hash function.

  This gives the user the ability to store more entries when a bucket is full, in comparison with the previous implementation.

  The API has not been changed, although new fields have been added in the `rte_hash` structure, which has been changed to internal use only.

  The main change when creating a new table is that the number of entries per bucket is now fixed, so its parameter is ignored now (it is still there to maintain the same parameters structure).

  Also, the maximum burst size in lookup_burst function hash been increased to 64, to improve performance.

- **Optimized KNI RX burst size computation.**

  Optimized KNI RX burst size computation by avoiding checking how many entries are in `kni->rx_q` prior to actually pulling them from the fifo.

- **Added KNI multicast.**

  Enabled adding multicast addresses to KNI interfaces by adding an empty callback for `set_rx_mode` (typically used for setting up hardware) so that the ioctl succeeds. This is the same thing as the Linux tap interface does.

- **Added cmdline polling mode.**

  Added the ability to process console input in the same thread as packet processing by using the `poll()` function.

- **Added VXLAN Tunnel End point sample application.**

  Added a Tunnel End point (TEP) sample application that simulates a VXLAN Tunnel Endpoint (VTEP) termination in DPDK. It is used to demonstrate the offload and filtering capabilities of Intel XL710 10/40 GbE NICsfor VXLAN packets.

- **Enabled combining of the ‘‘-m‘‘ and ‘‘–no-huge‘‘ EAL options.**

  Added option to allow combining of the `-m` and `--no-huge` EAL command line options.

This allows user application to run as non-root but with higher memory allocations, and removes a constraint on `--no-huge` mode being limited to 64M.

2.2 Resolved Issues

- **acl: Fix ambiguity between test rules.**

  Some test rules had equal priority for the same category. That could cause an ambiguity in building the trie and test results.

- **acl: Fix invalid rule wildness calculation for bitmask field type.**
- **acl: Fix matching rule.**
- **acl: Fix unneeded trie splitting for subset of rules.**

  When rebuilding a trie for limited rule-set, don't try to split the rule-set even further.

- **app/testpmd: Fix crash when port id out of bound.**

  Fixed issues in testpmd where using a port greater than 32 would cause a seg fault.

  Fixes: edab33b1c01d ("app/testpmd: support port hotplug")

- **app/testpmd: Fix reply to a multicast ICMP request.**

  Set the IP source and destination addresses in the IP header of the ICMP reply.

- **app/testpmd: fix MAC address in ARP reply.**

  Fixed issue where in the `icmpecho` forwarding mode, ARP replies from testpmd contain invalid zero-filled MAC addresses.

  Fixes: 31db4d38de72 ("net: change arp header struct declaration")

- **app/testpmd: fix default flow control values.**

  Fixes: 422a20a4e62d ("app/testpmd: fix uninitialized flow control variables")

- **bonding: Fix crash when stopping inactive slave.**
- **bonding: Fix device initialization error handling.**
- **bonding: Fix initial link status of slave.**

  On Fortville NIC, link status change interrupt callback was not executed when slave in bonding was (re-)started.

- **bonding: Fix socket id for LACP slave.**

  Fixes: 46fb43683679 ("bond: add mode 4")

- **bonding: Fix device initialization error handling.**
- **cmdline: Fix small memory leak.**

  A function in `cmdline.c` had a return that did not free the buf properly.

- **config: Enable same drivers options for Linux and BSD.**

  Enabled vector ixgbe and i40e bulk alloc for BSD as it is already done for Linux.

Fixes: 304caba12643 (“config: fix bsd options”) Fixes: 0ff3324da2eb (“ixgbe: rework vector pmd following mbuf changes”)

- **devargs: Fix crash on failure.**

  This problem occurred when passing an invalid PCI id to the blacklist API in devargs.

- **e1000/i40e: Fix descriptor done flag with odd address.**

- **e1000/igb: fix ieee1588 timestamping initialization.**

  Fixed issue with e1000 ieee1588 timestamp initialization. On initialization the IEEE1588 functions read the system time to set their timestamp. However, on some 1G NICs, for example, i350, system time is disabled by default and the IEEE1588 timestamp was always 0.

- **eal/bsd: Fix inappropriate header guards.**

- **eal/bsd: Fix virtio on FreeBSD.**

  Closing the `/dev/io` fd caused a SIGBUS in inb/outb instructions as the process lost the IOPL privileges once the fd is closed.

  Fixes: 8a312224bcde (“eal/bsd: fix fd leak”)

- **eal/linux: Fix comments on vfio MSI.**

- **eal/linux: Fix irq handling with igb_uio.**

  Fixed an issue where the the introduction of `uio_pci_generic` broke interrupt handling with igb_uio.

  Fixes: c112df6875a5 (“eal/linux: toggle interrupt for uio_pci_generic”)

- **eal/linux: Fix numa node detection.**

- **eal/linux: Fix socket value for undetermined numa node.**

  Sets zero as the default value of pci device numa_node if the socket could not be determined. This provides the same default value as FreeBSD which has no NUMA support, and makes the return value of `rte_eth_dev_socket_id()` be consistent with the API description.

- **eal/ppc: Fix cpu cycle count for little endian.**

  On IBM POWER8 PPC64 little endian architecture, the definition of tsc union will be different. This fix enables the right output from `rte_rdtsc()`.

- **ethdev: Fix check of threshold for TX freeing.**

  Fixed issue where the parameter to `tx_free_thresh` was not consistent between the drivers.

- **ethdev: Fix crash if malloc of user callback fails.**

  If `rte_zmalloc()` failed in `rte_eth_dev_callback_register` then the NULL pointer would be dereferenced.

- **ethdev: Fix illegal port access.**

  To obtain a detachable flag, `pci_drv` is accessed in `rte_eth_dev_is_detachable()`. However `pci_drv` is only valid if port is enabled. Fixed by checking `rte_eth_dev_is_valid_port()` first.

- **ethdev: Make tables const.**
- **ethdev: Rename and extend the mirror type.**
- **examples/distributor: Fix debug macro.**

  The macro to turn on additional debug output when the app was compiled with `-DDEBUG` was broken.

  Fixes: 07db4a975094 ("examples/distributor: new sample app")
- **examples/kni: Fix crash on exit.**
- **examples/vhost: Fix build with debug enabled.**

  Fixes: 72ec8d77ac68 ("examples/vhost: rework duplicated code")
- **fm10k: Fix RETA table initialization.**

  The fm10k driver has 128 RETA entries in 32 registers, but it only initialized the first 32 when doing multiple RX queue configurations. This fix initializes all 128 entries.
- **fm10k: Fix RX buffer size.**
- **fm10k: Fix TX multi-segment frame.**
- **fm10k: Fix TX queue cleaning after start error.**
- **fm10k: Fix Tx queue cleaning after start error.**
- **fm10k: Fix default mac/vlan in switch.**
- **fm10k: Fix interrupt fault handling.**
- **fm10k: Fix jumbo frame issue.**
- **fm10k: Fix mac/vlan filtering.**
- **fm10k: Fix maximum VF number.**
- **fm10k: Fix maximum queue number for VF.**

  Both PF and VF shared code in function `fm10k_stats_get()`. The function worked with PF, but had problems with VF since it has less queues than PF.

  Fixes: a6061d9e7075 ("fm10k: register PF driver")
- **fm10k: Fix queue disabling.**
- **fm10k: Fix switch synchronization.**
- **i40e/base: Fix error handling of NVM state update.**
- **i40e/base: Fix hardware port number for pass-through.**
- **i40e/base: Rework virtual address retrieval for lan queue.**
- **i40e/base: Update LED blinking.**
- **i40e/base: Workaround for PHY type with firmware < 4.4.**
- **i40e: Disable setting of PHY configuration.**
- **i40e: Fix SCTP flow director.**

- **i40e: Fix check of descriptor done flag.**

  Fixes: 4861cde46116 ("i40e: new poll mode driver") Fixes: 05999aab4ca6 ("i40e: add or delete flow director")

- **i40e: Fix condition to get VMDQ info.**
- **i40e: Fix registers access from big endian CPU.**
- **i40evf: Clear command when error occurs.**
- **i40evf: Fix RSS with less RX queues than TX queues.**
- **i40evf: Fix crash when setup TX queues.**
- **i40evf: Fix jumbo frame support.**
- **i40evf: Fix offload capability flags.**

  Added checksum offload capability flags which have already been supported for a long time.

- **ivshmem: Fix crash in corner case.**

  Fixed issues where depending on the configured segments it was possible to hit a segmentation fault as a result of decrementing an unsigned index with value 0.

  Fixes: 40b966a211ab ("ivshmem: library changes for mmaping using ivshmem")

- **ixgbe/base: Fix SFP probing.**
- **ixgbe/base: Fix TX pending clearing.**
- **ixgbe/base: Fix X550 CS4227 address.**
- **ixgbe/base: Fix X550 PCIe master disabling.**
- **ixgbe/base: Fix X550 check.**
- **ixgbe/base: Fix X550 init early return.**
- **ixgbe/base: Fix X550 link speed.**
- **ixgbe/base: Fix X550em CS4227 speed mode.**
- **ixgbe/base: Fix X550em SFP+ link stability.**
- **ixgbe/base: Fix X550em UniPHY link configuration.**
- **ixgbe/base: Fix X550em flow control for KR backplane.**
- **ixgbe/base: Fix X550em flow control to be KR only.**
- **ixgbe/base: Fix X550em link setup without SFP.**
- **ixgbe/base: Fix X550em mux after MAC reset.**

  Fixes: d2e72774e58c ("ixgbe/base: support X550")

- **ixgbe/base: Fix bus type overwrite.**
- **ixgbe/base: Fix init handling of X550em link down.**
- **ixgbe/base: Fix lan id before first i2c access.**
- **ixgbe/base: Fix mac type checks.**

- **ixgbe/base: Fix tunneled UDP and TCP frames in flow director.**
- **ixgbe: Check mbuf refcnt when clearing a ring.**

  The function to clear the TX ring when a port was being closed, e.g. on exit in testpmd, was not checking the mbuf refcnt before freeing it. Since the function in the vector driver to clear the ring after TX does not setting the pointer to NULL post-free, this caused crashes if mbuf debugging was turned on.
- **ixgbe: Fix RX with buffer address not word aligned.**

  Niantic HW expects the Header Buffer Address in the RXD must be word aligned.
- **ixgbe: Fix RX with buffer address not word aligned.**
- **ixgbe: Fix Rx queue reset.**

  Fix to reset vector related RX queue fields to their initial values.

  Fixes: c95584dc2b18 (“ixgbe: new vectorized functions for Rx/Tx”)
- **ixgbe: Fix TSO in IPv6.**

  When TSO was used with IPv6, the generated frames were incorrect. The L4 frame was OK, but the length field of IPv6 header was not populated correctly.
- **ixgbe: Fix X550 flow director check.**
- **ixgbe: Fix check for split packets.**

  The check for split packets to be reassembled in the vector ixgbe PMD was incorrectly only checking the first 16 elements of the array instead of all 32.

  Fixes: cf4b4708a88a (“ixgbe: improve slow-path perf with vector scattered Rx”)
- **ixgbe: Fix data access on big endian cpu.**
- **ixgbe: Fix flow director flexbytes offset.**

  Fixes: d54a9888267c (“ixgbe: support flexpayload configuration of flow director”)
- **ixgbe: Fix number of segments with vector scattered Rx.**

  Fixes: cf4b4708a88a (ixgbe: improve slow-path perf with vector scattered Rx)
- **ixgbe: Fix offload config option name.**

  The RX_OLFLAGS option was renamed from DISABLE to ENABLE in the driver code and Linux config. It is now renamed also in the BSD config and documentation.

  Fixes: 359f106a69a9 (“ixgbe: prefer enabling olflags rather than not disabling”)
- **ixgbe: Fix release queue mbufs.**

  The calculations of what mbufs were valid in the RX and TX queues were incorrect when freeing the mbufs for the vector PMD. This led to crashes due to invalid reference counts when mbuf debugging was turned on, and possibly other more subtle problems (such as mbufs being freed when in use) in other cases.

  Fixes: c95584dc2b18 (“ixgbe: new vectorized functions for Rx/Tx”)
- **ixgbe: Move PMD specific fields out of base driver.**

  Move `rx_bulk_alloc_allowed` and `rx_vec_allowed` from `ixgbe_hw` to `ixgbe_adapter`.

Fixes: 01fa1d6215fa ("ixgbe: unify Rx setup")

- **ixgbe: Rename TX queue release function.**
- **ixgbevf: Fix RX function selection.**

  The logic to select ixgbe the VF RX function is different than the PF.

- **ixgbevf: Fix link status for PF up/down events.**
- **kni: Fix RX loop limit.**

  Loop processing packets dequeued from rx_q was using the number of packets requested, not how many it actually received.

- **kni: Fix ioctl in containers, like Docker.**
- **kni: Fix multicast ioctl handling.**
- **log: Fix crash after log_history dump.**
- **lpm: Fix big endian support.**
- **lpm: Fix depth small entry add.**
- **mbuf: Fix cloning with private mbuf data.**

  Added a new `priv_size` field in mbuf structure that should be initialized at mbuf pool creation. This field contains the size of the application private data in mbufs.

  Introduced new static inline functions `rte_mbuf_from_indirect()` and `rte_mbuf_to_baddr()` to replace the existing macros, which take the private size into account when attaching and detaching mbufs.

- **mbuf: Fix data room size calculation in pool init.**

  Deduct the mbuf data room size from `mempool->elt_size` and `priv_size`, instead of using an hardcoded value that is not related to the real buffer size.

  To use `rte_pktmbuf_pool_init()`, the user can either:

  – Give a NULL parameter to rte_pktmbuf_pool_init(): in this case, the private size is assumed to be 0, and the room size is `mp->elt_size - sizeof(struct rte_mbuf)`.

  – Give the `rte_pktmbuf_pool_private` filled with appropriate data_room_size and priv_size values.

- **mbuf: Fix init when private size is not zero.**

  Allow the user to use the default `rte_pktmbuf_init()` function even if the mbuf private size is not 0.

- **mempool: Add structure for object headers.**

  Each object stored in mempools are prefixed by a header, allowing for instance to retrieve the mempool pointer from the object. When debug is enabled, a cookie is also added in this header that helps to detect corruptions and double-frees.

  Introduced a structure that materializes the content of this header, and will simplify future patches adding things in this header.

- **mempool: Fix pages computation to determine number of objects.**

- **mempool: Fix returned value after counting objects.**

  Fixes: 148f963fb532 ("xen: core library changes")

- **mlx4: Avoid requesting TX completion events to improve performance.**

  Instead of requesting a completion event for each TX burst, request it on a fixed schedule once every MLX4_PMD_TX_PER_COMP_REQ (currently 64) packets to improve performance.

- **mlx4: Fix compilation as a shared library and on 32 bit platforms.**

- **mlx4: Fix possible crash on scattered mbuf allocation failure.**

  Fixes issue where failing to allocate a segment, `mlx4_rx_burst_sp()` could call `rte_pktmbuf_free()` on an incomplete scattered mbuf whose next pointer in the last segment is not set.

- **mlx4: Fix support for multiple vlan filters.**

  This fixes the "Multiple RX VLAN filters can be configured, but only the first one works" bug.

- **pcap: Fix storage of name and type in queues.**

  pcap_rx_queue/pcap_tx_queue should store it's own copy of name/type values, not the pointer to temporary allocated space.

- **pci: Fix memory leaks and needless increment of map address.**

- **pci: Fix uio mapping differences between linux and bsd.**

- **port: Fix unaligned access to metadata.**

  Fix RTE_MBUF_METADATA macros to allow for unaligned accesses to meta-data fields.

- **ring: Fix return of new port id on creation.**

- **timer: Fix race condition.**

  Eliminate problematic race condition in `rte_timer_manage()` that can lead to corruption of per-lcore pending-lists (implemented as skip-lists).

- **vfio: Fix overflow of BAR region offset and size.**

  Fixes: 90a1633b2347 ("eal/Linux: allow to map BARs with MSI-X tables")

- **vhost: Fix enqueue/dequeue to handle chained vring descriptors.**

- **vhost: Fix race for connection fd.**

- **vhost: Fix virtio freeze due to missed interrupt.**

- **virtio: Fix crash if CQ is not negotiated.**

  Fix NULL dereference if virtio control queue is not negotiated.

- **virtio: Fix ring size negotiation.**

  Negotiate the virtio ring size. The host may allow for very large rings but application may only want a smaller ring. Conversely, if the number of descriptors requested exceeds the virtio host queue size, then just silently use the smaller host size.

  This fixes issues with virtio in non-QEMU environments. For example Google Compute Engine allows up to 16K elements in ring.

- **vmxnet3: Fix link state handling.**

2.3 Known Issues

- When running the `vmdq` sample or `vhost` sample applications with the Intel(R) XL710 (i40e) NIC, the configuration option `CONFIG_RTE_MAX_QUEUES_PER_PORT` should be increased from 256 to 1024.
- VM power manager may not work on systems with more than 64 cores.

2.4 API Changes

- The order that user supplied RX and TX callbacks are called in has been changed to the order that they were added (fifo) in line with end-user expectations. The previous calling order was the reverse of this (lifo) and was counter intuitive for users. The actual API is unchanged.

2.5 ABI Changes

- The `rte_hash` structure has been changed to internal use only.

CHAPTER

THREE

DPDK RELEASE 2.0

3.1 New Features

- Poll-mode driver support for an early release of the PCIE host interface of the Intel(R) Ethernet Switch FM10000.
  - Basic Rx/Tx functions for PF/VF
  - Interrupt handling support for PF/VF
  - Per queue start/stop functions for PF/VF
  - Support Mailbox handling between PF/VF and PF/Switch Manager
  - Receive Side Scaling (RSS) for PF/VF
  - Scatter receive function for PF/VF
  - Reta update/query for PF/VF
  - VLAN filter set for PF
  - Link status query for PF/VF

Note: The software is intended to run on pre-release hardware and may contain unknown or unresolved defects or issues related to functionality and performance. The poll mode driver is also pre-release and will be updated to a released version post hardware and base driver release. Should the official hardware release be made between DPDK releases an updated poll-mode driver will be made available.

- Link Bonding
  - Support for adaptive load balancing (mode 6) to the link bonding library.
  - Support for registration of link status change callbacks with link bonding devices.
  - Support for slaves devices which do not support link status change interrupts in the link bonding library via a link status polling mechanism.
- PCI Hotplug with NULL PMD sample application
- ABI versioning
- x32 ABI
- Non-EAL Thread Support
- Multi-pthread Support

- Re-order Library
- ACL for AVX2
- Architecture Independent CRC Hash
- uio_pci_generic Support
- KNI Optimizations
- Vhost-user support
- Virtio (link, vlan, mac, port IO, perf)
- IXGBE-VF RSS
- RX/TX Callbacks
- Unified Flow Types
- Indirect Attached MBUF Flag
- Use default port configuration in TestPMD
- Tunnel offloading in TestPMD
- Poll Mode Driver - 40 GbE Controllers (librte_pmd_i40e)
  - Support for Flow Director
  - Support for ethertype filter
  - Support RSS in VF
  - Support configuring redirection table with different size from 1GbE and 10 GbE
  - 128/512 entries of 40GbE PF
  - 64 entries of 40GbE VF
  - Support configuring hash functions
  - Support for VXLAN packet on Intel® 40GbE Controllers
- Poll Mode Driver for Mellanox ConnectX-3 EN adapters (mlx4)

Note: This PMD is only available for Linux and is disabled by default due to external dependencies (libibverbs and libmlx4). Please refer to the NIC drivers guide for more information.

- Packet Distributor Sample Application
- Job Stats library and Sample Application.
- Enhanced Jenkins hash (jhash) library

Note: The hash values returned by the new jhash library are different from the ones returned by the previous library.

CHAPTER

FOUR

DPDK RELEASE 1.8

4.1 New Features

- Link Bonding
  - Support for 802.3ad link aggregation (mode 4) and transmit load balancing (mode 5) to the link bonding library.
  - Support for registration of link status change callbacks with link bonding devices.
  - Support for slaves devices which do not support link status change interrupts in the link bonding library via a link status polling mechanism.
- Poll Mode Driver - 40 GbE Controllers (librte_pmd_i40e)
  - Support for Flow Director
  - Support for ethertype filter
  - Support RSS in VF
  - Support configuring redirection table with different size from 1GbE and 10 GbE
  - 128/512 entries of 40GbE PF
  - 64 entries of 40GbE VF
  - Support configuring hash functions
  - Support for VXLAN packet on IntelÂ® 40GbE Controllers
- Packet Distributor Sample Application

CHAPTER

FIVE

SUPPORTED OPERATING SYSTEMS

The following Linux distributions were successfully used to compiler or run DPDK.

- FreeBSD 10
- Fedora release 20
- Ubuntu 14.04 LTS
- Wind River Linux 6
- Red Hat Enterprise Linux 6.5
- SUSE Enterprise Linux 11 SP3

These distributions may need additional packages that are not installed by default, or a specific kernel. Refer to the Linux guide and FreeBSD guide for details.

CHAPTER

SIX

KNOWN ISSUES AND LIMITATIONS IN LEGACY RELEASES

This section describes known issues with the DPDK software that aren't covered in the version specific release notes sections.

6.1 Unit Test for Link Bonding may fail at test_tlb_tx_burst()

Description: Unit tests will fail in `test_tlb_tx_burst()` function with error for uneven distribution of packets.

Implication: Unit test link_bonding_autotest will fail.

Resolution/Workaround: There is no workaround available.

Affected Environment/Platform: Fedora 20.

Driver/Module: Link Bonding.

6.2 Pause Frame Forwarding does not work properly on igb

Description: For igb devices rte_eth_flow_ctrl_set does not work as expected. Pause frames are always forwarded on igb, regardless of the `RFCE`, `MPMCF` and `DPF` registers.

Implication: Pause frames will never be rejected by the host on 1G NICs and they will always be forwarded.

Resolution/Workaround: There is no workaround available.

Affected Environment/Platform: All.

Driver/Module: Poll Mode Driver (PMD).

6.3 In packets provided by the PMD, some flags are missing

Description: In packets provided by the PMD, some flags are missing. The application does not have access to information provided by the hardware (packet is broadcast, packet is multicast, packet is IPv4 and so on).

Implication: The `ol_flags` field in the `rte_mbuf` structure is not correct and should not be used.

Resolution/Workaround: The application has to parse the Ethernet header itself to get the information, which is slower.

Affected Environment/Platform: All.

Driver/Module: Poll Mode Driver (PMD).

6.4 The rte_malloc library is not fully implemented

Description: The `rte_malloc` library is not fully implemented.

Implication: All debugging features of rte_malloc library described in architecture documentation are not yet implemented.

Resolution/Workaround: No workaround available.

Affected Environment/Platform: All.

Driver/Module: `rte_malloc`.

6.5 HPET reading is slow

Description: Reading the HPET chip is slow.

Implication: An application that calls `rte_get_hpet_cycles()` or `rte_timer_manage()` runs slower.

Resolution/Workaround: The application should not call these functions too often in the main loop. An alternative is to use the TSC register through `rte_rdtsc()` which is faster, but specific to an lcore and is a cycle reference, not a time reference.

Affected Environment/Platform: All.

Driver/Module: Environment Abstraction Layer (EAL).

6.6 HPET timers do not work on the Osage customer reference platform

Description: HPET timers do not work on the Osage customer reference platform which includes an Intel® Xeon® processor 5500 series processor) using the released BIOS from Intel.

Implication: On Osage boards, the implementation of the `rte_delay_us()` function must be changed to not use the HPET timer.

Resolution/Workaround: This can be addressed by building the system with the `CONFIG_RTE_LIBEAL_USE_HPET=n` configuration option or by using the `--no-hpet` EAL option.

Affected Environment/Platform: The Osage customer reference platform. Other vendor platforms with Intel® Xeon® processor 5500 series processors should work correctly, provided the BIOS supports HPET.

Driver/Module: `lib/librte_eal/common/include/rte_cycles.h`

6.7 Not all variants of supported NIC types have been used in testing

Description: The supported network interface cards can come in a number of variants with different device ID's. Not all of these variants have been tested with the Intel® DPDK.

The NIC device identifiers used during testing:

- Intel® Ethernet Controller XL710 for 40GbE QSFP+ [8086:1584]
- Intel® Ethernet Controller XL710 for 40GbE QSFP+ [8086:1583]
- Intel® Ethernet Controller X710 for 10GbE SFP+ [8086:1572]
- Intel® 82576 Gigabit Ethernet Controller [8086:10c9]
- Intel® 82576 Quad Copper Gigabit Ethernet Controller [8086:10e8]
- Intel® 82580 Dual Copper Gigabit Ethernet Controller [8086:150e]
- Intel® I350 Quad Copper Gigabit Ethernet Controller [8086:1521]
- Intel® 82599 Dual Fibre 10 Gigabit Ethernet Controller [8086:10fb]
- Intel® Ethernet Server Adapter X520-T2 [8086: 151c]
- Intel® Ethernet Controller X540-T2 [8086:1528]
- Intel® 82574L Gigabit Network Connection [8086:10d3]
- Emulated Intel® 82540EM Gigabit Ethernet Controller [8086:100e]
- Emulated Intel® 82545EM Gigabit Ethernet Controller [8086:100f]
- Intel® Ethernet Server Adapter X520-4 [8086:154a]
- Intel® Ethernet Controller I210 [8086:1533]

Implication: Risk of issues with untested variants.

Resolution/Workaround: Use tested NIC variants. For those supported Ethernet controllers, additional device IDs may be added to the software if required.

Affected Environment/Platform: All.

Driver/Module: Poll-mode drivers

6.8 Multi-process sample app requires exact memory mapping

Description: The multi-process example application assumes that it is possible to map the hugepage memory to the same virtual addresses in client and server applications. Occasionally, very rarely with 64-bit, this does not occur and a client application will fail on startup. The Linux "address-space layout randomization" security feature can sometimes cause this to occur.

Implication: A multi-process client application fails to initialize.

Resolution/Workaround: See the "Multi-process Limitations" section in the Intel® DPDK Programmer's Guide for more information.

Affected Environment/Platform: All.

Driver/Module: Multi-process example application

6.9 Packets are not sent by the 1 GbE/10 GbE SR-IOV driver when the source MAC is not the MAC assigned to the VF NIC

Description: The 1 GbE/10 GbE SR-IOV driver can only send packets when the Ethernet header's source MAC address is the same as that of the VF NIC. The reason for this is that the Linux `ixgbe` driver module in the host OS has its anti-spoofing feature enabled.

Implication: Packets sent using the 1 GbE/10 GbE SR-IOV driver must have the source MAC address correctly set to that of the VF NIC. Packets with other source address values are dropped by the NIC if the application attempts to transmit them.

Resolution/Workaround: Configure the Ethernet source address in each packet to match that of the VF NIC.

Affected Environment/Platform: All.

Driver/Module: 1 GbE/10 GbE VF Poll Mode Driver (PMD).

6.10 SR-IOV drivers do not fully implement the rte_ethdev API

Description: The SR-IOV drivers only supports the following rte_ethdev API functions:

- rte_eth_dev_configure()
- rte_eth_tx_queue_setup()
- rte_eth_rx_queue_setup()
- rte_eth_dev_info_get()
- rte_eth_dev_start()
- rte_eth_tx_burst()
- rte_eth_rx_burst()
- rte_eth_dev_stop()
- rte_eth_stats_get()
- rte_eth_stats_reset()
- rte_eth_link_get()
- rte_eth_link_get_no_wait()

Implication: Calling an unsupported function will result in an application error.

Resolution/Workaround: Do not use other rte_ethdev API functions in applications that use the SR-IOV drivers.

Affected Environment/Platform: All.

Driver/Module: VF Poll Mode Driver (PMD).

6.11 PMD does not work with –no-huge EAL command line parameter

Description: Currently, the DPDK does not store any information about memory allocated by `malloc()‘ (for example, NUMA node, physical address), hence PMD drivers do not work when the ‘‘--no-huge` command line parameter is supplied to EAL.

Implication: Sending and receiving data with PMD will not work.

Resolution/Workaround: Use huge page memory or use VFIO to map devices.

Affected Environment/Platform: Systems running the DPDK on Linux

Driver/Module: Poll Mode Driver (PMD).

6.12 Some hardware off-load functions are not supported by the VF Driver

Description: Currently, configuration of the following items is not supported by the VF driver:

- IP/UDP/TCP checksum offload
- Jumbo Frame Receipt
- HW Strip CRC

Implication: Any configuration for these items in the VF register will be ignored. The behavior is dependent on the current PF setting.

Resolution/Workaround: For the PF (Physical Function) status on which the VF driver depends, there is an option item under PMD in the config file. For others, the VF will keep the same behavior as PF setting.

Affected Environment/Platform: All.

Driver/Module: VF (SR-IOV) Poll Mode Driver (PMD).

6.13 Kernel crash on IGB port unbinding

Description: Kernel crash may occur when unbinding 1G ports from the igb_uio driver, on 2.6.3x kernels such as shipped with Fedora 14.

Implication: Kernel crash occurs.

Resolution/Workaround: Use newer kernels or do not unbind ports.

Affected Environment/Platform: 2.6.3x kernels such as shipped with Fedora 14

Driver/Module: IGB Poll Mode Driver (PMD).

6.14 Twinpond and Ironpond NICs do not report link status correctly

Description: Twin Pond/Iron Pond NICs do not bring the physical link down when shutting down the port.

Implication: The link is reported as up even after issuing `shutdown` command unless the cable is physically disconnected.

Resolution/Workaround: None.

Affected Environment/Platform: Twin Pond and Iron Pond NICs

Driver/Module: Poll Mode Driver (PMD).

6.15 Discrepancies between statistics reported by different NICs

Description: Gigabit Ethernet devices from Intel include CRC bytes when calculating packet reception statistics regardless of hardware CRC stripping state, while 10-Gigabit Ethernet devices from Intel do so only when hardware CRC stripping is disabled.

Implication: There may be a discrepancy in how different NICs display packet reception statistics.

Resolution/Workaround: None

Affected Environment/Platform: All.

Driver/Module: Poll Mode Driver (PMD).

6.16 Error reported opening files on DPDK initialization

Description: On DPDK application startup, errors may be reported when opening files as part of the initialization process. This occurs if a large number, for example, 500 or more, or if hugepages are used, due to the per-process limit on the number of open files.

Implication: The DPDK application may fail to run.

Resolution/Workaround: If using 2 MB hugepages, consider switching to a fewer number of 1 GB pages. Alternatively, use the `ulimit` command to increase the number of files which can be opened by a process.

Affected Environment/Platform: All.

Driver/Module: Environment Abstraction Layer (EAL).

6.17 Intel® QuickAssist Technology sample application does not work on a 32-bit OS on Shumway

Description: The Intel® Communications Chipset 89xx Series device does not fully support NUMA on a 32-bit OS. Consequently, the sample application cannot work properly on Shumway, since it requires NUMA on both nodes.

Implication: The sample application cannot work in 32-bit mode with emulated NUMA, on multi-socket boards.

Resolution/Workaround: There is no workaround available.

Affected Environment/Platform: Shumway

Driver/Module: All.

6.18 IEEE1588 support possibly not working with an Intel® Ethernet Controller I210 NIC

Description: IEEE1588 support is not working with an Intel® Ethernet Controller I210 NIC.

Implication: IEEE1588 packets are not forwarded correctly by the Intel® Ethernet Controller I210 NIC.

Resolution/Workaround: There is no workaround available.

Affected Environment/Platform: All.

Driver/Module: IGB Poll Mode Driver

6.19 Differences in how different Intel NICs handle maximum packet length for jumbo frame

Description: 10 Gigabit Ethernet devices from Intel do not take VLAN tags into account when calculating packet size while Gigabit Ethernet devices do so for jumbo frames.

Implication: When receiving packets with VLAN tags, the actual maximum size of useful payload that Intel Gigabit Ethernet devices are able to receive is 4 bytes (or 8 bytes in the case of packets with extended VLAN tags) less than that of Intel 10 Gigabit Ethernet devices.

Resolution/Workaround: Increase the configured maximum packet size when using Intel Gigabit Ethernet devices.

Affected Environment/Platform: All.

Driver/Module: Poll Mode Driver (PMD).

6.20 Binding PCI devices to igb_uio fails on Linux kernel 3.9 when more than one device is used

Description: A known bug in the uio driver included in Linux kernel version 3.9 prevents more than one PCI device to be bound to the igb_uio driver.

Implication: The Poll Mode Driver (PMD) will crash on initialization.

Resolution/Workaround: Use earlier or later kernel versions, or apply the following patch.

Affected Environment/Platform: Linux systems with kernel version 3.9

Driver/Module: igb_uio module

6.21 GCC might generate Intel® AVX instructions for processors without Intel® AVX support

Description: When compiling Intel® DPDK (and any DPDK app), gcc may generate Intel® AVX instructions, even when the processor does not support Intel® AVX.

Implication: Any DPDK app might crash while starting up.

Resolution/Workaround: Either compile using icc or set `EXTRA_CFLAGS='-03'` prior to compilation.

Affected Environment/Platform: Platforms which processor does not support Intel® AVX.

Driver/Module: Environment Abstraction Layer (EAL).

6.22 Ethertype filter could receive other packets (non-assigned) in Niantic

Description: On Intel® Ethernet Controller 82599EB When Ethertype filter (priority enable) was set, unmatched packets also could be received on the assigned queue, such as ARP packets without 802.1q tags or with the user priority not equal to set value. Launch the testpmd by disabling RSS and with multiply queues, then add the ethertype filter like the following and then start forwarding:

```
add_ethertype_filter 0 ethertype 0x0806 priority enable 3 queue 2 index 1
```

When sending ARP packets without 802.1q tag and with user priority as non-3 by tester, all the ARP packets can be received on the assigned queue.

Implication: The user priority comparing in Ethertype filter cannot work probably. It is a NIC's issue due to the following: "In fact, ETQF.UP is not functional, and the information will be added in errata of 82599 and X540."

Resolution/Workaround: None

Affected Environment/Platform: All.

Driver/Module: Poll Mode Driver (PMD).

6.23 Cannot set link speed on Intel® 40G Ethernet controller

Description: On Intel® 40G Ethernet Controller you cannot set the link to specific speed.

Implication: The link speed cannot be changed forcibly, though it can be configured by application.

Resolution/Workaround: None

Affected Environment/Platform: All.

Driver/Module: Poll Mode Driver (PMD).

6.24 Stopping the port does not down the link on Intel® 40G Ethernet controller

Description: On Intel® 40G Ethernet Controller stopping the port does not really down the port link.

Implication: The port link will be still up after stopping the port.

Resolution/Workaround: None

Affected Environment/Platform: All.

Driver/Module: Poll Mode Driver (PMD).

6.25 Devices bound to igb_uio with VT-d enabled do not work on Linux kernel 3.15-3.17

Description: When VT-d is enabled (`iommu=pt intel_iommu=on`), devices are 1:1 mapped. In the Linux kernel unbinding devices from drivers removes that mapping which result in IOMMU errors. Introduced in Linux kernel 3.15 commit, solved in Linux kernel 3.18 commit.

Implication: Devices will not be allowed to access memory, resulting in following kernel errors:

```
dmar: DRHD: handling fault status reg 2
dmar: DMAR:[DMA Read] Request device [02:00.0] fault addr a0c58000
DMAR:[fault reason 02] Present bit in context entry is clear
```

Resolution/Workaround: Use earlier or later kernel versions, or avoid driver binding on boot by blacklisting the driver modules. I.e., in the case of `ixgbe`, we can pass the kernel command line option: `modprobe.blacklist=ixgbe`. This way we do not need to unbind the device to bind it to igb_uio.

Affected Environment/Platform: Linux systems with kernel versions 3.15 to 3.17.

Driver/Module: `igb_uio` module.

6.26 VM power manager may not work on systems with more than 64 cores

Description: When using VM power manager on a system with more than 64 cores, VM(s) should not use cores 64 or higher.

Implication: VM power manager should not be used with VM(s) that are using cores 64 or above.

Resolution/Workaround: Do not use cores 64 or above.

Affected Environment/Platform: Platforms with more than 64 cores.

Driver/Module: VM power manager application.

CHAPTER

SEVEN

ABI AND API DEPRECATION

See the `guidelines document for details of the ABI policy`. API and ABI deprecation notices are to be posted here.

7.1 Deprecation Notices

- Significant ABI changes are planned for struct rte_eth_dev to support up to 1024 queues per port. This change will be in release 2.2. There is no backward compatibility planned from release 2.2. All binaries will need to be rebuilt from release 2.2.
- ABI changes are planned for struct rte_intr_handle, struct rte_eth_conf and struct eth_dev_ops to support interrupt mode feature from release 2.1. Those changes may be enabled in the release 2.1 with CONFIG_RTE_NEXT_ABI.
- The EAL function rte_eal_pci_close_one is deprecated because renamed to rte_eal_pci_detach.
- The Macros RTE_HASH_BUCKET_ENTRIES_MAX and RTE_HASH_KEY_LENGTH_MAX are deprecated and will be removed with version 2.2.
- The function rte_jhash2 is deprecated and should be removed.
- The field mem_location of the rte_lpm structure is deprecated and should be removed as well as the macros RTE_LPM_HEAP and RTE_LPM_MEMZONE.
- Significant ABI changes are planned for struct rte_mbuf, struct rte_kni_mbuf, and several PKT_RX_ flags will be removed, to support unified packet type from release 2.1. Those changes may be enabled in the upcoming release 2.1 with CONFIG_RTE_NEXT_ABI.
- librte_malloc library has been integrated into librte_eal. The 2.1 release creates a dummy/empty malloc library to fulfill binaries with dynamic linking dependencies on librte_malloc.so. Such dummy library will not be created from release 2.2 so binaries will need to be rebuilt.
- The following fields have been deprecated in rte_eth_stats: imissed, ibadcrc, ibadlen, imcasts, fdirmatch, fdirmiss, tx_pause_xon, rx_pause_xon, tx_pause_xoff, rx_pause_xoff
- API for flow director filters has been replaced by rte_eth_dev_filter_ctrl. Following old API is deprecated and will be removed with version 2.2 without backward compatibility. Functions: rte_eth_dev_fdir_*. Structures: rte_fdir_*, rte_eth_fdir. Enums: rte_l4type, rte_iptype.

- ABI changes are planned for struct rte_eth_fdir_flow_ext in order to support flow director filtering in VF. The release 2.1 does not contain these ABI changes, but release 2.2 will, and no backwards compatibility is planned.
- ABI change is planned to extend the SCTP flow's key input from release 2.1. The change may be enabled in the release 2.1 with CONFIG_RTE_NEXT_ABI.
- ABI changes are planned for struct rte_eth_fdir_filter and rte_eth_fdir_masks in order to support new flow director modes, MAC VLAN and Cloud, on x550. The MAC VLAN mode means the MAC and VLAN are monitored. The Cloud mode is for VxLAN and NVGRE, and the tunnel type, TNI/VNI, inner MAC and inner VLAN are monitored. The release 2.2 will contain these changes without backwards compatibility.
- librte_kni: Functions based on port id are deprecated for a long time and should be removed (rte_kni_create, rte_kni_get_port_id and rte_kni_info_get).
- librte_pmd_ring: The deprecated functions rte_eth_ring_pair_create and rte_eth_ring_pair_attach should be removed.
- ABI changes are planned for struct virtio_net in order to support vhost-user multiple queues feature. It should be integrated in release 2.2 without backward compatibility.
- The scheduler hierarchy structure (rte_sched_port_hierarchy) will change to allow for a larger number of subport entries. The number of available traffic_classes and queues may also change. The mbuf structure element for sched hierarchy will also change from a single 32 bit to a 64 bit structure.
- The scheduler statistics structure will change to allow keeping track of RED actions.
- librte_acl: The structure rte_acl_ipv4vlan_rule is deprecated and should be removed as well as the associated functions rte_acl_ipv4vlan_add_rules and rte_acl_ipv4vlan_build.
- librte_cfgfile: In order to allow for longer names and values, the value of macros CFG_NAME_LEN and CFG_NAME_VAL will be increased. Most likely, the new values will be 64 and 256, respectively.
- librte_port: Macros to access the packet meta-data stored within the packet buffer will be adjusted to cover the packet mbuf structure as well, as currently they are able to access any packet buffer location except the packet mbuf structure.
- librte_table LPM: A new parameter to hold the table name will be added to the LPM table parameter structure.
- librte_table: New functions for table entry bulk add/delete will be added to the table operations structure.
- librte_table hash: Key mask parameter will be added to the hash table parameter structure for 8-byte key and 16-byte key extendible bucket and LRU tables.
- librte_pipeline: The prototype for the pipeline input port, output port and table action handlers will be updated: the pipeline parameter will be added, the packets mask parameter will be either removed (for input port action handler) or made input-only.