DPDK

DATA PLANE DEVELOPMENT KIT

Release Notes

Release 16.11.11

Nov 15, 2019

CONTENTS

CHAPTER ONE

DESCRIPTION OF RELEASE

This document contains the release notes for Data Plane Development Kit (DPDK) release version 16.11.11 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 16.11

2.1 New Features

- **Added software parser for packet type.**
  - Added a new function `rte_pktmbuf_read()` to read the packet data from an mbuf chain, linearizing if required.
  - Added a new function `rte_net_get_ptype()` to parse an Ethernet packet in an mbuf chain and retrieve its packet type from software.
  - Added new functions `rte_get_ptype_*()` to dump a packet type as a string.
- **Improved offloads support in mbuf.**
  - Added a new function `rte_raw_cksum_mbuf()` to process the checksum of data embedded in an mbuf chain.
  - Added new Rx checksum flags in mbufs to describe more states: unknown, good, bad, or not present (useful for virtual drivers). This modification was done for IP and L4.
  - Added a new Rx LRO mbuf flag, used when packets are coalesced. This flag indicates that the segment size of original packets is known.
- **Added vhost-user dequeue zero copy support.**

  The copy in the dequeue path is avoided in order to improve the performance. In the VM2VM case, the boost is quite impressive. The bigger the packet size, the bigger performance boost you may get. However, for the VM2NIC case, there are some limitations, so the boost is not as impressive as the VM2VM case. It may even drop quite a bit for small packets.

  For that reason, this feature is disabled by default. It can be enabled when the `RTE_VHOST_USER_DEQUEUE_ZERO_COPY` flag is set. Check the VHost section of the Programming Guide for more information.
- **Added vhost-user indirect descriptors support.**

  If the indirect descriptor feature is enabled, each packet sent by the guest will take exactly one slot in the enqueue virtqueue. Without this feature, as in the current version, even 64 bytes packets take two slots with Virtio PMD on guest side.

  The main impact is better performance for 0% packet loss use-cases, as it behaves as if the virtqueue size was enlarged, so more packets can be buffered in the case of system perturbations. On the downside, small performance degradations were measured when running micro-benchmarks.

- **Added vhost PMD xstats.**

  Added extended statistics to vhost PMD from a per port perspective.

- **Supported offloads with virtio.**

  Added support for the following offloads in virtio:

  - Rx/Tx checksums.
  - LRO.
  - TSO.

- **Added virtio NEON support for ARM.**

  Added NEON support for ARM based virtio.

- **Updated the ixgbe base driver.**

  Updated the ixgbe base driver, including the following changes:

  - Added X550em_a 10G PHY support.
  - Added support for flow control auto negotiation for X550em_a 1G PHY.
  - Added X550em_a FW ALEF support.
  - Increased mailbox version to `ixgbe_mbox_api_13`.
  - Added two MAC operations for Hyper-V support.

- **Added APIs for VF management to the ixgbe PMD.**

  Eight new APIs have been added to the ixgbe PMD for VF management from the PF. The declarations for the API's can be found in `rte_pmd_ixgbe.h`.

- **Updated the enic driver.**

  - Added update to use interrupt for link status checking instead of polling.
  - Added more flow director modes on UCS Blade with firmware version >= 2.0(13e).
  - Added full support for MTU update.
  - Added support for the `rte_eth_rx_queue_count` function.

- **Updated the mlx5 driver.**

  - Added support for RSS hash results.
  - Added several performance improvements.
  - Added several bug fixes.

- **Updated the QAT PMD.**

  The QAT PMD was updated with additional support for:

  - MD5_HMAC algorithm.
  - SHA224-HMAC algorithm.
  - SHA384-HMAC algorithm.

  - GMAC algorithm.
  - KASUMI (F8 and F9) algorithm.
  - 3DES algorithm.
  - NULL algorithm.
  - C3XXX device.
  - C62XX device.

- **Added openssl PMD.**

  A new crypto PMD has been added, which provides several ciphering and hashing algorithms. All cryptography operations use the Openssl library crypto API.

- **Updated the IPsec example.**

  Updated the IPsec example with the following support:

  - Configuration file support.
  - AES CBC IV generation with cipher forward function.
  - AES GCM/CTR mode.

- **Added support for new gcc -march option.**

  The GCC 4.9 `-march` option supports the Intel processor code names. The config option `RTE_MACHINE` can be used to pass code names to the compiler via the `-march` flag.

2.2 Resolved Issues

2.2.1 Drivers

- **enic: Fixed several flow director issues.**
- **enic: Fixed inadvertent setting of L4 checksum ptype on ICMP packets.**
- **enic: Fixed high driver overhead when servicing Rx queues beyond the first.**

2.3 Known Issues

- **L3fwd-power app does not work properly when Rx vector is enabled.**

  The L3fwd-power app doesn’t work properly with some drivers in vector mode since the queue monitoring works differently between scalar and vector modes leading to incorrect frequency scaling. In addition, L3fwd-power application requires the mbuf to have correct packet type set but in some drivers the vector mode must be disabled for this.

  Therefore, in order to use L3fwd-power, vector mode should be disabled via the config file.

- **Digest address must be supplied for crypto auth operation on QAT PMD.**

  The cryptodev API specifies that if the rte_crypto_sym_op.digest.data field, and by inference the digest.phys_addr field which points to the same location, is not set for an auth operation the driver is to understand that the digest result is located immediately following the region over which the digest is computed. The QAT PMD doesn't correctly handle this case and reads and writes to an incorrect location.

Callers can workaround this by always supplying the digest virtual and physical address fields in the rte_crypto_sym_op for an auth operation.

2.4 API Changes

- The driver naming convention has been changed to make them more consistent. It especially impacts `--vdev` arguments. For example `eth_pcap` becomes `net_pcap` and `cryptodev_aesni_mb_pmd` becomes `crypto_aesni_mb`.

  For backward compatibility an alias feature has been enabled to support the original names.

- The log history has been removed.
- The `rte_ivshmem` feature (including library and EAL code) has been removed in 16.11 because it had some design issues which were not planned to be fixed.
- The `file_name` data type of `struct rte_port_source_params` and `struct rte_port_sink_params` is changed from `char *` to `const char *`.
- **Improved device/driver hierarchy and generalized hotplugging.**

  The device and driver relationship has been restructured by introducing generic classes. This paves the way for having PCI, VDEV and other device types as instantiated objects rather than classes in themselves. Hotplugging has also been generalized into EAL so that Ethernet or crypto devices can use the common infrastructure.

  - Removed `pmd_type` as a way of segregation of devices.
  - Moved `numa_node` and `devargs` into `rte_driver` from `rte_pci_driver`. These can now be used by any instantiated object of `rte_driver`.
  - Added `rte_device` class and all PCI and VDEV devices inherit from it
  - Renamed devinit/devuninit handlers to probe/remove to make it more semantically correct with respect to the device <=> driver relationship.
  - Moved hotplugging support to EAL. Hereafter, PCI and vdev can use the APIs `rte_eal_dev_attach` and `rte_eal_dev_detach`.
  - Renamed helpers and support macros to make them more synonymous with their device types (e.g. `PMD_REGISTER_DRIVER` => `RTE_PMD_REGISTER_PCI`).
  - Device naming functions have been generalized from ethdev and cryptodev to EAL. `rte_eal_pci_device_name` has been introduced for obtaining unique device name from PCI Domain-BDF description.
  - Virtual device registration APIs have been added: `rte_eal_vdrv_register` and `rte_eal_vdrv_unregister`.

2.5 ABI Changes

2.6 Shared Library Versions

The libraries prepended with a plus sign were incremented in this version.

```
  librte_acl.so.2
  librte_cfgfile.so.2
  librte_cmdline.so.2
+ librte_cryptodev.so.2
  librte_distributor.so.1
+ librte_eal.so.3
+ librte_ethdev.so.5
  librte_hash.so.2
  librte_ip_frag.so.1
  librte_jobstats.so.1
  librte_kni.so.2
  librte_kvargs.so.1
  librte_lpm.so.2
  librte_mbuf.so.2
  librte_mempool.so.2
  librte_meter.so.1
  librte_net.so.1
  librte_pdump.so.1
  librte_pipeline.so.3
  librte_pmd_bond.so.1
  librte_pmd_ring.so.2
  librte_port.so.3
  librte_power.so.1
  librte_reorder.so.1
  librte_ring.so.1
  librte_sched.so.1
  librte_table.so.2
  librte_timer.so.1
  librte_vhost.so.3
```

2.7 Tested Platforms

1. SuperMicro 1U
   - BIOS: 1.0c
   - Processor: Intel(R) Atom(TM) CPU C2758 @ 2.40GHz
2. SuperMicro 1U
   - BIOS: 1.0a
   - Processor: Intel(R) Xeon(R) CPU D-1540 @ 2.00GHz
   - Onboard NIC: Intel(R) X552/X557-AT (2x10G)
     - Firmware-version: 0x800001cf
     - Device ID (PF/VF): 8086:15ad /8086:15a8
   - kernel driver version: 4.2.5 (ixgbe)
3. SuperMicro 2U

  - BIOS: 1.0a
  - Processor: Intel(R) Xeon(R) CPU E5-4667 v3 @ 2.00GHz

4. Intel(R) Server board S2600GZ
   - BIOS: SE5C600.86B.02.02.0002.122320131210
   - Processor: Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz
5. Intel(R) Server board W2600CR
   - BIOS: SE5C600.86B.02.01.0002.082220131453
   - Processor: Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz
6. Intel(R) Server board S2600CWT
   - BIOS: SE5C610.86B.01.01.0009.060120151350
   - Processor: Intel(R) Xeon(R) CPU E5-2699 v3 @ 2.30GHz
7. Intel(R) Server board S2600WTT
   - BIOS: SE5C610.86B.01.01.0005.101720141054
   - Processor: Intel(R) Xeon(R) CPU E5-2699 v3 @ 2.30GHz
8. Intel(R) Server board S2600WTT
   - BIOS: SE5C610.86B.11.01.0044.090120151156
   - Processor: Intel(R) Xeon(R) CPU E5-2695 v4 @ 2.10GHz
9. Intel(R) Server board S2600WTT
   - Processor: Intel(R) Xeon(R) CPU E5-2697 v2 @ 2.70GHz
10. Intel(R) Server
    - Intel(R) Xeon(R) CPU E5-2697 v3 @ 2.60GHz
11. IBM(R) Power8(R)
    - Machine type-model: 8247-22L
    - Firmware FW810.21 (SV810_108)
    - Processor: POWER8E (raw), AltiVec supported

2.8 Tested NICs

1. Intel(R) Ethernet Controller X540-AT2
   - Firmware version: 0x80000389
   - Device id (pf): 8086:1528
   - Driver version: 3.23.2 (ixgbe)
2. Intel(R) 82599ES 10 Gigabit Ethernet Controller
   - Firmware version: 0x61bf0001
   - Device id (pf/vf): 8086:10fb / 8086:10ed

- Driver version: 4.0.1-k (ixgbe)

3. Intel(R) Corporation Ethernet Connection X552/X557-AT 10GBASE-T
   - Firmware version: 0x800001cf
   - Device id (pf/vf): 8086:15ad / 8086:15a8
   - Driver version: 4.2.5 (ixgbe)
4. Intel(R) Ethernet Converged Network Adapter X710-DA4 (4x10G)
   - Firmware version: 5.05
   - Device id (pf/vf): 8086:1572 / 8086:154c
   - Driver version: 1.5.23 (i40e)
5. Intel(R) Ethernet Converged Network Adapter X710-DA2 (2x10G)
   - Firmware version: 5.05
   - Device id (pf/vf): 8086:1572 / 8086:154c
   - Driver version: 1.5.23 (i40e)
6. Intel(R) Ethernet Converged Network Adapter XL710-QDA1 (1x40G)
   - Firmware version: 5.05
   - Device id (pf/vf): 8086:1584 / 8086:154c
   - Driver version: 1.5.23 (i40e)
7. Intel(R) Ethernet Converged Network Adapter XL710-QDA2 (2X40G)
   - Firmware version: 5.05
   - Device id (pf/vf): 8086:1583 / 8086:154c
   - Driver version: 1.5.23 (i40e)
8. Intel(R) Corporation I350 Gigabit Network Connection
   - Firmware version: 1.48, 0x800006e7
   - Device id (pf/vf): 8086:1521 / 8086:1520
   - Driver version: 5.2.13-k (igb)
9. Intel(R) Ethernet Multi-host Controller FM10000
   - Firmware version: N/A
   - Device id (pf/vf): 8086:15d0
   - Driver version: 0.17.0.9 (fm10k)
10. Mellanox(R) ConnectX(R)-4 10G MCX4111A-XCAT (1x10G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - MLNX_OFED: 3.4-1.0.0.0
    - Firmware version: 12.17.1010

11. Mellanox(R) ConnectX(R)-4 10G MCX4121A-XCAT (2x10G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - MLNX_OFED: 3.4-1.0.0.0
    - Firmware version: 12.17.1010
12. Mellanox(R) ConnectX(R)-4 25G MCX4111A-ACAT (1x25G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - MLNX_OFED: 3.4-1.0.0.0
    - Firmware version: 12.17.1010
13. Mellanox(R) ConnectX(R)-4 25G MCX4121A-ACAT (2x25G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - MLNX_OFED: 3.4-1.0.0.0
    - Firmware version: 12.17.1010
14. Mellanox(R) ConnectX(R)-4 40G MCX4131A-BCAT/MCX413A-BCAT (1x40G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - MLNX_OFED: 3.4-1.0.0.0
    - Firmware version: 12.17.1010
15. Mellanox(R) ConnectX(R)-4 40G MCX415A-BCAT (1x40G)
    - Host interface: PCI Express 3.0 x16
    - Device ID: 15b3:1013
    - MLNX_OFED: 3.4-1.0.0.0
    - Firmware version: 12.17.1010
16. Mellanox(R) ConnectX(R)-4 50G MCX4131A-GCAT/MCX413A-GCAT (1x50G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - MLNX_OFED: 3.4-1.0.0.0
    - Firmware version: 12.17.1010
17. Mellanox(R) ConnectX(R)-4 50G MCX414A-BCAT (2x50G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - MLNX_OFED: 3.4-1.0.0.0

   - Firmware version: 12.17.1010

18. Mellanox(R) ConnectX(R)-4 50G MCX415A-GCAT/MCX416A-BCAT/MCX416A-GCAT (2x50G)
    - Host interface: PCI Express 3.0 x16
    - Device ID: 15b3:1013
    - MLNX_OFED: 3.4-1.0.0.0
    - Firmware version: 12.17.1010
19. Mellanox(R) ConnectX(R)-4 50G MCX415A-CCAT (1x100G)
    - Host interface: PCI Express 3.0 x16
    - Device ID: 15b3:1013
    - MLNX_OFED: 3.4-1.0.0.0
    - Firmware version: 12.17.1010
20. Mellanox(R) ConnectX(R)-4 100G MCX416A-CCAT (2x100G)
    - Host interface: PCI Express 3.0 x16
    - Device ID: 15b3:1013
    - MLNX_OFED: 3.4-1.0.0.0
    - Firmware version: 12.17.1010
21. Mellanox(R) ConnectX(R)-4 Lx 10G MCX4121A-XCAT (2x10G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1015
    - MLNX_OFED: 3.4-1.0.0.0
    - Firmware version: 14.17.1010
22. Mellanox(R) ConnectX(R)-4 Lx 25G MCX4121A-ACAT (2x25G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1015
    - MLNX_OFED: 3.4-1.0.0.0
    - Firmware version: 14.17.1010

2.9 Tested OSes

- CentOS 7.2
- Fedora 23
- Fedora 24
- FreeBSD 10.3
- FreeBSD 11

- Red Hat Enterprise Linux Server release 6.7 (Santiago)
- Red Hat Enterprise Linux Server release 7.0 (Maipo)
- Red Hat Enterprise Linux Server release 7.2 (Maipo)
- SUSE Enterprise Linux 12
- Wind River Linux 6.0.0.26
- Wind River Linux 8
- Ubuntu 14.04
- Ubuntu 15.04
- Ubuntu 16.04

2.10 Fixes in 16.11 LTS Release

2.10.1 16.11.1

- app/test: fix symmetric session free in crypto perf tests
- app/testpmd: fix check for invalid ports
- app/testpmd: fix static build link ordering
- crypto/aesni_gcm: fix IV size in capabilities
- crypto/aesni_gcm: fix J0 padding bytes
- crypto/aesni_mb: fix incorrect crypto session
- crypto/openssl: fix extra bytes written at end of data
- crypto/openssl: fix indentation in guide
- crypto/qat: fix IV size in capabilities
- crypto/qat: fix to avoid buffer overwrite in OOP case
- cryptodev: fix crash on null dereference
- cryptodev: fix loop in device query
- devargs: reset driver name pointer on parsing failure
- drivers/crypto: fix different auth/cipher keys
- ethdev: check maximum number of queues for statistics
- ethdev: fix extended statistics name index
- ethdev: fix port data mismatched in multiple process model
- ethdev: fix port lookup if none
- ethdev: remove invalid function from version map
- examples/ethtool: fix driver information
- examples/ethtool: fix querying non-PCI devices

- examples/ip_pipeline: fix coremask limitation
- examples/ip_pipeline: fix parsing of pass-through pipeline
- examples/l2fwd-crypto: fix overflow
- examples/vhost: fix calculation of mbuf count
- examples/vhost: fix lcore initialization
- mempool: fix API documentation
- mempool: fix stack handler dequeue
- net/af_packet: fix fd use after free
- net/bnx2x: fix Rx mode configuration
- net/cxgbe/base: initialize variable before reading EEPROM
- net/cxgbe: fix parenthesis on bitwise operation
- net/ena: fix setting host attributes
- net/enic: fix hardcoding of some flow director masks
- net/enic: fix memory leak with oversized Tx packets
- net/enic: remove unnecessary function parameter attributes
- net/i40e: enable auto link update for 25G
- net/i40e: fix Rx checksum flag
- net/i40e: fix TC bandwidth definition
- net/i40e: fix VF reset flow
- net/i40e: fix checksum flag in x86 vector Rx
- net/i40e: fix crash in close
- net/i40e: fix deletion of all macvlan filters
- net/i40e: fix ethertype filter on X722
- net/i40e: fix link update delay
- net/i40e: fix logging for Tx free threshold check
- net/i40e: fix segment number in reassemble process
- net/i40e: fix wrong return value when handling PF message
- net/i40e: fix xstats value mapping
- net/i40evf: fix casting between structs
- net/i40evf: fix reporting of imissed packets
- net/ixgbe: fix blocked interrupts
- net/ixgbe: fix received packets number for ARM
- net/ixgbe: fix received packets number for ARM NEON
- net/ixgbevf: fix max packet length

- net/mlx5: fix RSS hash result for flows
- net/mlx5: fix Rx packet validation and type
- net/mlx5: fix Tx doorbell
- net/mlx5: fix endianness in Tx completion queue
- net/mlx5: fix inconsistent link status
- net/mlx5: fix leak when starvation occurs
- net/mlx5: fix link status query
- net/mlx5: fix memory leak when parsing device params
- net/mlx5: fix missing inline attributes
- net/mlx5: fix updating total length of multi-packet send
- net/mlx: fix IPv4 and IPv6 packet type
- net/nfp: fix VLAN offload flags check
- net/nfp: fix typo in Tx offload capabilities
- net/pcap: fix timestamps in output pcap file
- net/qede/base: fix FreeBSD build
- net/qede: add vendor/device id info
- net/qede: fix PF fastpath status block index
- net/qede: fix filtering code
- net/qede: fix function declaration
- net/qede: fix per queue statisitics
- net/qede: fix resource leak
- net/vhost: fix socket file deleted on stop
- net/vhost: fix unix socket not removed as closing
- net/virtio-user: fix not properly reset device
- net/virtio-user: fix wrongly get/set features
- net/virtio: fix build without virtio-user
- net/virtio: fix crash when number of virtio devices > 1
- net/virtio: fix multiple process support
- net/virtio: fix performance regression due to TSO
- net/virtio: fix rewriting LSC flag
- net/virtio: fix wrong Rx/Tx method for secondary process
- net/virtio: optimize header reset on any layout
- net/virtio: store IO port info locally
- net/virtio: store PCI operators pointer locally

- net/vmxnet3: fix Rx deadlock
- pci: fix check of mknod
- pmdinfogen: fix endianness with cross-compilation
- pmdinfogen: fix null dereference
- sched: fix crash when freeing port
- usertools: fix active interface detection when binding
- vdev: fix detaching with alias
- vfio: fix file descriptor leak in multi-process
- vhost: allow many vhost-user ports
- vhost: do not GSO when no header is present
- vhost: fix dead loop in enqueue path
- vhost: fix guest/host physical address mapping
- vhost: fix long stall of negotiation
- vhost: fix memory leak

2.10.2 16.11.2

- app/testpmd: fix TC mapping in DCB init config
- app/testpmd: fix crash at mbuf pool creation
- app/testpmd: fix exit without freeing resources
- app/testpmd: fix init config for multi-queue mode
- app/testpmd: fix number of mbufs in pool
- app: enable HW CRC strip by default
- crypto/openssl: fix AAD capabilities for AES-GCM
- crypto/openssl: fix AES-GCM capability
- crypto/qat: fix AES-GCM authentication length
- crypto/qat: fix IV zero physical address
- crypto/qat: fix dequeue statistics
- cryptodev: fix API digest length comments
- doc: add limitation of AAD size to QAT guide
- doc: explain zlib dependency for bnx2x
- eal/linux: fix build with glibc 2.25
- eal: fix debug macro redefinition
- examples/ip_fragmentation: fix check of packet type
- examples/l2fwd-crypto: fix AEAD tests when AAD is zero

- examples/l2fwd-crypto: fix packets array index
- examples/l2fwd-crypto: fix padding calculation
- examples/l3fwd-power: fix Rx descriptor size
- examples/l3fwd-power: fix handling no Rx queue
- examples/load_balancer: fix Tx flush
- examples/multi_process: fix timer update
- examples/performance-thread: fix build on FreeBSD
- examples/performance-thread: fix build on FreeBSD 10.0
- examples/performance-thread: fix compilation on Suse 11 SP2
- examples/quota_watermark: fix requirement for 2M pages
- examples: enable HW CRC strip by default
- examples: fix build clean on FreeBSD
- kni: fix build with kernel 4.11
- kni: fix crash caused by freeing mempool
- kni: fix possible memory leak
- mbuf: fix missing includes in exported header
- mk: fix lib filtering when linking app
- mk: fix quoting for ARM mtune argument
- mk: fix shell errors when building with clang
- net/bnx2x: fix transmit queue free threshold
- net/bonding: allow configuring jumbo frames without slaves
- net/bonding: fix updating slave link status
- net/cxgbe: fix possible null pointer dereference
- net/e1000/base: fix multicast setting in VF
- net/ena: cleanup if refilling of Rx descriptors fails
- net/ena: fix Rx descriptors allocation
- net/ena: fix delayed cleanup of Rx descriptors
- net/ena: fix return of hash control flushing
- net/fm10k: fix memory overflow in 32-bit SSE Rx
- net/fm10k: fix pointer cast
- net/i40e/base: fix potential out of bound array access
- net/i40e: add missing 25G link speed
- net/i40e: ensure vector mode is not used with QinQ
- net/i40e: fix TC bitmap of VEB

- net/i40e: fix VF link speed
- net/i40e: fix VF link status update
- net/i40e: fix allocation check
- net/i40e: fix compile error
- net/i40e: fix hash input set on X722
- net/i40e: fix incorrect packet index reference
- net/i40e: fix mbuf alloc failed counter
- net/i40e: fix memory overflow in 32-bit SSE Rx
- net/i40e: fix setup when bulk is disabled
- net/igb: fix VF MAC address setting
- net/igb: fix VF MAC address setting
- net/ixgbe/base: fix build error
- net/ixgbe: fix Rx queue blocking issue
- net/ixgbe: fix TC bandwidth setting
- net/ixgbe: fix VF Rx mode for allmulticast disabled
- net/ixgbe: fix all queues drop setting of DCB
- net/ixgbe: fix memory overflow in 32-bit SSE Rx
- net/ixgbe: fix multi-queue mode check in SRIOV mode
- net/ixgbe: fix setting MTU on stopped device
- net/ixgbevf: set xstats id values
- net/mlx4: fix Rx after mbuf alloc failure
- net/mlx4: fix returned values upon failed probing
- net/mlx4: update link status upon probing with LSC
- net/mlx5: fix Tx when first segment size is too short
- net/mlx5: fix VLAN stripping indication
- net/mlx5: fix an uninitialized variable
- net/mlx5: fix returned values upon failed probing
- net/mlx5: fix reusing Rx/Tx queues
- net/mlx5: fix supported packets types
- net/nfp: fix packet/data length conversion
- net/pcap: fix using mbuf after freeing it
- net/qede/base: fix find zero bit macro
- net/qede: fix FW version string for VF
- net/qede: fix default MAC address handling

- net/qede: fix fastpath rings reset phase
- net/qede: fix missing UDP protocol in RSS offload types
- net/thunderx: fix 32-bit build
- net/thunderx: fix build on FreeBSD
- net/thunderx: fix deadlock in Rx path
- net/thunderx: fix stats access out of bounds
- net/virtio-user: fix address on 32-bit system
- net/virtio-user: fix overflow
- net/virtio: disable LSC interrupt if MSIX not enabled
- net/virtio: fix MSI-X for modern devices
- net/virtio: fix crash when closing twice
- net/virtio: fix link status always being up
- net/virtio: fix link status always down
- net/virtio: fix queue notify
- net/vmxnet3: fix build with gcc 7
- net/vmxnet3: fix queue size changes
- net: fix stripped VLAN flag for offload emulation
- nic_uio: fix device binding at boot
- pci: fix device registration on FreeBSD
- test/cmdline: fix missing break in switch
- test/mempool: free mempool on exit
- test: enable HW CRC strip by default
- vfio: fix disabling INTx
- vfio: fix secondary process start
- vhost: change log levels in client mode
- vhost: fix dequeue zero copy
- vhost: fix false sharing
- vhost: fix fd leaks for vhost-user server mode
- vhost: fix max queues
- vhost: fix multiple queue not enabled for old kernels
- vhost: fix use after free

2.10.3 16.11.3

- contigmem: do not zero pages during each mmap
- contigmem: free allocated memory on error
- crypto/aesni_mb: fix HMAC supported key sizes
- cryptodev: fix device stop function
- crypto/openssl: fix HMAC supported key sizes
- crypto/qat: fix HMAC supported key sizes
- crypto/qat: fix NULL authentication hang
- crypto/qat: fix SHA384-HMAC block size
- doc: remove incorrect limitation on AESNI-MB PMD
- doc: remove incorrect limitation on QAT PMD
- eal: fix config file path when checking process
- examples/l2fwd-crypto: fix application help
- examples/l2fwd-crypto: fix option parsing
- examples/l2fwd-crypto: fix padding
- examples/l3fwd: fix IPv6 packet type parse
- examples/qos_sched: fix build for less lcores
- ip_frag: free mbufs on reassembly table destroy
- kni: fix build with gcc 7.1
- lpm: fix index of tbl8
- mbuf: fix debug checks for headroom and tailroom
- mbuf: fix doxygen comment of bulk alloc
- mbuf: fix VXLAN port in comment
- mem: fix malloc element resize with padding
- net/bnxt: check invalid L2 filter id
- net/bnxt: enable default VNIC allocation
- net/bnxt: fix autoneg on 10GBase-T links
- net/bnxt: fix get link config
- net/bnxt: fix reporting of link status
- net/bnxt: fix set link config
- net/bnxt: fix set link config
- net/bnxt: fix vnic cleanup
- net/bnxt: free filter before reusing it
- net/bonding: change link status check to no-wait

- net/bonding: fix number of bonding Tx/Rx queues
- net/bonding: fix when NTT flag updated
- net/cxgbe: fix port statistics
- net/e1000: fix LSC interrupt
- net/ena: fix cleanup of the Tx bufs
- net/enic: fix build with gcc 7.1
- net/enic: fix crash when freeing 0 packet to mempool
- net/fm10k: initialize link status in device start
- net/i40e: add return value checks
- net/i40e/base: fix Tx error stats on VF
- net/i40e: exclude internal packet's byte count
- net/i40e: fix division by 0
- net/i40e: fix ethertype filter for new FW
- net/i40e: fix link down and negotiation
- net/i40e: fix Rx data segment buffer length
- net/i40e: fix VF statistics
- net/igb: fix add/delete of flex filters
- net/igb: fix checksum valid flags
- net/igb: fix flex filter length
- net/ixgbe: fix mirror rule index overflow
- net/ixgbe: fix Rx/Tx queue interrupt for x550 devices
- net/mlx4: fix mbuf poisoning in debug code
- net/mlx4: fix probe failure report
- net/mlx5: fix build with gcc 7.1
- net/mlx5: fix completion buffer size
- net/mlx5: fix exception handling
- net/mlx5: fix inconsistent link status query
- net/mlx5: fix redundant free of Tx buffer
- net/qede: fix chip details print
- net/virtio: do not claim to support LRO
- net/virtio: do not falsely claim to do IP checksum
- net/virtio-user: fix crash when detaching device
- net/virtio: zero the whole memory zone
- net/vmxnet3: fix filtering on promiscuous disabling

- net/vmxnet3: fix receive queue memory leak
- Revert "ip_frag: free mbufs on reassembly table destroy"
- test/bonding: fix memory corruptions
- test/bonding: fix mode 4 names
- test/bonding: fix namespace of the RSS tests
- test/bonding: fix parameters of a balance Tx
- test/crypto: fix overflow
- test/crypto: fix wrong AAD setting
- vhost: fix checking of device features
- vhost: fix guest pages memory leak
- vhost: fix IP checksum
- vhost: fix TCP checksum
- vhost: make page logging atomic

2.10.4 16.11.4

- app/testpmd: fix forwarding between non consecutive ports
- app/testpmd: fix invalid port id parameters
- app/testpmd: fix mapping of user priority to DCB TC
- app/testpmd: fix packet throughput after stats reset
- app/testpmd: fix RSS structure initialisation
- app/testpmd: fix topology error message
- buildtools: check allocation error in pmdinfogen
- buildtools: fix icc build
- cmdline: fix compilation with -Og
- cmdline: fix warning for unused return value
- config: fix bnx2x option for armv7a
- cryptodev: fix build with -Ofast
- crypto/qat: fix SHA512-HMAC supported key size
- drivers/crypto: use snprintf return value correctly
- eal/bsd: fix missing interrupt stub functions
- eal: copy raw strings taken from command line
- eal: fix auxv open check for ARM and PPC
- eal/x86: fix atomic cmpset
- examples/ipsec-secgw: fix IPv6 payload length

- examples/ipsec-secgw: fix IP version check
- examples/l2fwd-cat: fix build with PQOS 1.4
- examples/l2fwd-crypto: fix uninitialized errno value
- examples/l2fwd_fork: fix message pool init
- examples/l3fwd-acl: check fseek return
- examples/multi_process: fix received message length
- examples/performance-thread: check thread creation
- examples/performance-thread: fix out-of-bounds sched array
- examples/performance-thread: fix out-of-bounds tls array
- examples/qos_sched: fix uninitialized config
- hash: fix eviction counter
- kni: fix build on RHEL 7.4
- kni: fix build on SLE12 SP3
- kni: fix ethtool build with kernel 4.11
- lpm6: fix compilation with -Og
- mem: fix malloc element free in debug mode
- net/bnxt: fix a bit shift operation
- net/bnxt: fix an issue with broadcast traffic
- net/bnxt: fix a potential null pointer dereference
- net/bnxt: fix interrupt handler
- net/bnxt: fix link handling and configuration
- net/bnxt: fix Rx offload capability
- net/bnxt: fix Tx offload capability
- net/bnxt: set checksum offload flags correctly
- net/bnxt: update status of Rx IP/L4 CKSUM
- net/bonding: fix LACP slave deactivate behavioral
- net/cxgbe: fix memory leak
- net/enic: fix assignment
- net/enic: fix packet loss after MTU change
- net/enic: fix possible null pointer dereference
- net: fix inner L2 length in packet type parser
- net/i40e/base: fix bool definition
- net/i40e: fix clear xstats bug in VF
- net/i40e: fix flexible payload configuration

- net/i40e: fix flow control watermark mismatch
- net/i40e: fix i40evf MAC filter table
- net/i40e: fix mbuf free in vector Tx
- net/i40e: fix memory leak if VF init fails
- net/i40e: fix mirror rule reset when port is closed
- net/i40e: fix mirror with firmware 6.0
- net/i40e: fix packet count for PF
- net/i40e: fix PF notify issue when VF is not up
- net/i40e: fix Rx packets number for NEON
- net/i40e: fix Rx queue interrupt mapping in VF
- net/i40e: fix uninitialized variable
- net/i40e: fix variable assignment
- net/i40e: fix VF cannot forward packets issue
- net/i40e: fix VFIO interrupt mapping in VF
- net/igb: fix memcpy length
- net/igb: fix Rx interrupt with VFIO and MSI-X
- net/ixgbe: fix adding a mirror rule
- net/ixgbe: fix mapping of user priority to TC
- net/ixgbe: fix PF DCB info
- net/ixgbe: fix uninitialized variable
- net/ixgbe: fix VFIO interrupt mapping in VF
- net/ixgbe: fix VF RX hang
- net/mlx5: fix clang build
- net/mlx5: fix clang compilation error
- net/mlx5: fix link speed bitmasks
- net/mlx5: fix probe failure report
- net/mlx5: fix Tx stats error counter definition
- net/mlx5: fix Tx stats error counter logic
- net/mlx5: improve stack usage during link update
- net/nfp: fix RSS
- net/nfp: fix stats struct initial value
- net/pcap: fix memory leak in dumper open
- net/qede/base: fix API return types
- net/qede/base: fix division by zero

- net/qede/base: fix for VF malicious indication
- net/qede/base: fix macros to check chip revision/metal
- net/qede/base: fix number of app table entries
- net/qede/base: fix return code to align with FW
- net/qede/base: fix to use a passed ptt handle
- net/qede: fix icc build
- net/virtio: fix compilation with -Og
- net/virtio: fix mbuf port for simple Rx function
- net/virtio: fix queue setup consistency
- net/virtio: fix Tx packet length stats
- net/virtio: fix untrusted scalar value
- net/virtio: flush Rx queues on start
- net/vmxnet3: fix dereference before null check
- net/vmxnet3: fix MAC address set
- net/vmxnet3: fix memory leak when releasing queues
- pdump: fix possible mbuf leak on failure
- ring: guarantee load/load order in enqueue and dequeue
- test: fix assignment operation
- test/memzone: fix memory leak
- test/pmd_perf: fix crash with multiple devices
- timer: use 64-bit specific code on more platforms
- uio: fix compilation with -Og
- usertools: fix device binding with python 3
- vfio: fix close unchecked file descriptor

2.10.5 16.11.5

- app/procinfo: add compilation option in config
- app/testpmd: fix crash of txonly with multiple segments
- app/testpmd: fix flow director filter
- app/testpmd: fix port index in RSS forward config
- app/testpmd: fix port topology in RSS forward config
- bus/pci: fix interrupt handler type
- contigmem: fix build on FreeBSD 12
- crypto/qat: fix allocation check and leak

- crypto/qat: fix null auth algo overwrite
- doc: fix outdated link to IPsec white paper
- eal/ppc: remove the braces in memory barrier macros
- eal/ppc: support sPAPR IOMMU for vfio-pci
- eal: update assertion macro
- eal/x86: use lock-prefixed instructions for SMP barrier
- ethdev: fix data alignment
- ethdev: fix link autonegotiation value
- ethdev: fix missing imissed counter in xstats
- ethdev: fix typo in functions comment
- examples/bond: check mbuf allocation
- examples/exception_path: align stats on cache line
- examples/ip_pipeline: fix timer period unit
- examples/ipsec-secgw: fix corner case for SPI value
- examples/l3fwd-power: fix frequency detection
- examples/l3fwd-power: fix Rx without interrupt
- examples/vhost: fix sending ARP packet to self
- examples/vhost: fix startup check
- igb_uio: fix IRQ disable on recent kernels
- igb_uio: fix MSI-X IRQ assignment with new IRQ function
- igb_uio: switch to new irq function for MSI-X
- keepalive: fix state alignment
- kni: fix build with kernel 4.15
- lpm: fix ARM big endian build
- malloc: fix end for bounded elements
- malloc: protect stats with lock
- mbuf: cleanup function to get last segment
- mbuf: fix NULL freeing when debug enabled
- mem: fix mmap error check on huge page attach
- memzone: fix leak on allocation error
- mk: fix external build
- mk: support renamed Makefile in external project
- net/bnxt: fix broadcast cofiguration
- net/bnxt: fix group info usage

- net/bnxt: fix headroom initialization
- net/bnxt: fix link speed setting with autoneg off
- net/bnxt: fix Rx checksum flags
- net/bnxt: fix size of Tx ring in HW
- net/bnxt: parse checksum offload flags
- net/bnxt: support new PCI IDs
- net/bonding: check error of MAC address setting
- net/bonding: fix activated slave in 8023ad mode
- net/bonding: fix setting slave MAC addresses
- net/e1000: fix mailbox interrupt handler
- net/e1000: fix VF Rx interrupt enabling
- net/ena: do not set Tx L4 offloads in Rx path
- net/enic: fix crash due to static max number of queues
- net/fm10k: fix logical port delete
- net/i40e: add debug logs when writing global registers
- net/i40e: add warnings when writing global registers
- net/i40e/base: fix compile issue for GCC 6.3
- net/i40e/base: fix link LED blink
- net/i40e/base: fix NVM lock
- net/i40e: check multi-driver option parsing
- net/i40e: fix ARM big endian build
- net/i40e: fix flag for MAC address write
- net/i40e: fix flow director Rx resource defect
- net/i40e: fix interrupt conflict when using multi-driver
- net/i40e: fix multiple driver support issue
- net/i40e: fix Rx interrupt
- net/i40e: fix VF reset stats crash
- net/i40e: fix VF Rx interrupt enabling
- net/i40e: fix VLAN offload setting
- net/i40e: fix VSI MAC filter on primary address change
- net/i40e: implement vector PMD for altivec
- net/igb: fix Tx queue number assignment
- net/ixgbe/base: add media type of fixed fiber
- net/ixgbe: fix ARM big endian build

- net/ixgbe: fix mailbox interrupt handler
- net/ixgbe: fix max queue number for VF
- net/ixgbe: fix reset error handling
- net/ixgbe: fix the failure of number of Tx queue check
- net/ixgbe: fix VF Rx interrupt enabling
- net/ixgbe: improve link state check on VF
- net/mlx5: fix deadlock of link status alarm
- net/mlx5: fix missing RSS capability
- net/mlx5: fix MTU update
- net/nfp: fix CRC strip check behaviour
- net/nfp: fix jumbo settings
- net/nfp: fix MTU settings
- net/pcap: fix the NUMA id display in logs
- net/qede/base: fix VF LRO tunnel configuration
- net/qede: fix clearing of queue stats
- net/qede: fix few log messages
- net/qede: fix MTU set and max Rx pkt len usage
- net/qede: fix to reject config with no Rx queue
- net/szedata2: fix check of mmap return value
- net/thunderx: fix multi segment Tx function return
- net/vhost: fix log messages on create/destroy
- net/virtio: fix incorrect cast
- net/virtio: fix mbuf data offset for simple Rx
- net/virtio: fix memory leak when reinitializing device
- net/virtio: fix queue flushing with vector Rx enabled
- net/virtio: fix resuming port with Rx vector path
- net/virtio: fix Rx and Tx handler selection for ARM32
- net/virtio: fix typo in function name
- net/virtio: fix vector Rx flushing
- net/virtio-user: fix start with kernel vhost
- pdump: fix error check when creating/canceling thread
- pmdinfogen: fix cross compilation for ARM big endian
- test/crypto: fix missing include
- test/memzone: fix freeing test

- test/memzone: fix NULL freeing
- test/memzone: fix wrong test
- test/memzone: handle previously allocated memzones
- test/pmd_perf: declare variables as static
- test: register test as failed if setup failed
- test/reorder: fix memory leak
- test/ring_perf: fix memory leak
- test/table: fix memory leak
- test/table: fix uninitialized parameter
- test/timer_perf: fix memory leak
- usertools/devbind: remove unused function
- vfio: fix enabled check on error
- vhost: do not take lock on owner reset
- vhost: fix crash
- vhost: fix dequeue zero copy with virtio1
- vhost: fix error code check when creating thread
- vhost: fix mbuf free
- vhost: protect active rings from async ring changes

2.10.6 16.11.6

- vhost: add support for non-contiguous indirect descs tables (fixes CVE-2018-1059)
- vhost: check all range is mapped when translating GPAs (fixes CVE-2018-1059)
- vhost: ensure all range is mapped when translating QVAs (fixes CVE-2018-1059)
- vhost: handle virtually non-contiguous buffers in Rx (fixes CVE-2018-1059)
- vhost: handle virtually non-contiguous buffers in Rx-mrg (fixes CVE-2018-1059)
- vhost: handle virtually non-contiguous buffers in Tx (fixes CVE-2018-1059)
- vhost-user: fix deadlock in case of NUMA realloc

2.10.7 16.11.7

- app/crypto-perf: fix parameters copy
- app/testpmd: fix burst stats reporting
- app/testpmd: fix command token
- app/testpmd: fix forward ports Rx flush
- app/testpmd: fix forward ports update

- app/testpmd: fix slave port detection
- app/testpmd: fix synchronic port hotplug
- app/testpmd: fix valid ports prints
- bus/pci: fix size of driver name buffer
- crypto/zuc: do not set default op status
- crypto/zuc: remove unnecessary check
- doc: fix a typo in the EAL guide
- drivers/net: fix icc deprecated parameter warning
- drivers/net: fix link autoneg value for virtual PMDs
- eal: declare trace buffer at top of own block
- eal: explicit cast in rwlock functions
- eal: explicit cast of builtin for bsf32
- eal: explicit cast of core id when getting index
- eal: fix casts in random functions
- eal: fix typo in doc of pointer offset macro
- eal/ppc: remove braces in SMP memory barrier macro
- eal: remove unused path pattern
- eal: support strlcpy function
- ethdev: explicit cast of buffered Tx number
- ethdev: explicit cast of queue count return
- ethdev: fix queue start
- ethdev: fix string length in name comparison
- ethdev: fix type and scope of variables in Rx burst
- ethdev: improve doc for name by port ID API
- examples/exception_path: limit core count to 64
- examples/performance-thread: fix return type of threads
- hash: explicit casts for truncation in CRC32c
- hash: fix comment for lookup
- hash: move stack declaration at top of CRC32c function
- ip_frag: fix double free of chained mbufs
- ip_frag: fix some debug logs
- kni: fix build on RHEL 7.5
- kvargs: fix syntax in comments
- mbuf: avoid integer promotion in prepend/adj/chain

- mbuf: explicit cast of headroom on reset
- mbuf: explicit cast of size on detach
- mbuf: explicit casts of reference counter
- mbuf: fix reference counter integer promotion
- mbuf: fix Tx checksum offload API doc
- mbuf: fix type of private size in detach
- mempool: fix leak when no objects are populated
- mempool: fix virtual address population
- memzone: fix size on reserving biggest memzone
- net/bnx2x: do not cast function pointers as a policy
- net/bnx2x: fix for PCI FLR after ungraceful exit
- net/bnx2x: fix KR2 device check
- net/bnx2x: fix memzone name overrun
- net/bnxt: avoid freeing memzone multiple times
- net/bnxt: fix endianness of flag
- net/bnxt: fix mbuf data offset initialization
- net/bnxt: fix Rx checksum flags
- net/bnxt: fix Rx checksum flags for tunnel frames
- net/bnxt: fix Rx drop setting
- net/bnxt: fix Rx mbuf and agg ring leak in dev stop
- net/bonding: clear started state if start fails
- net/bonding: export mode 4 slave info routine
- net/bonding: fix setting VLAN ID on slave ports
- net/enic: allocate stats DMA buffer upfront during probe
- net/enic: fix crash on MTU update with non-setup queues
- net: explicit cast in L4 checksum
- net: explicit cast of IP checksum to 16-bit
- net: explicit cast of multicast bit clearing
- net: explicit cast of protocol in IPv6 checksum
- net/i40e: fix failing to disable FDIR Tx queue
- net/i40e: fix intr callback unregister by adding retry
- net/i40e: fix link status update
- net/i40e: fix link update no wait
- net/i40e: fix shifts of signed values

- net/ixgbe: fix DCB configuration
- net/ixgbe: fix intr callback unregister by adding retry
- net/ixgbe: fix too many interrupts
- net/mlx5: fix ARM build
- net/mlx5: fix double free on error handling
- net/mlx5: fix resource leak in case of error
- net: move stack variable at top of VLAN strip function
- net/nfp: fix assigning port id in mbuf
- net/nfp: fix barrier location
- net/nfp: fix mbufs releasing when stop or close
- net/nfp: fix memcpy out of source range
- net/qede: fix alloc from socket 0
- net/qede: fix strncpy
- net/qede: fix unicast filter routine return code
- net/qede: replace strncpy by strlcpy
- net/szedata2: fix format string for PCI address
- net/szedata2: fix total stats
- net/thunderx: fix MTU configuration for jumbo packets
- net/vhost: initialise device as inactive
- net/virtio-user: fix hugepage files enumeration
- net/vmxnet3: keep link state consistent
- net/vmxnet3: set the queue shared buffer at start
- Revert “vhost: fix device cleanup at stop”
- spinlock/x86: move stack declaration before code
- test/distributor: fix return type of thread function
- test: fix memory flags test for low NUMA nodes number
- test/mempool: fix autotest retry
- test/pipeline: fix return type of stub miss
- test/pipeline: fix type of table entry parameter
- test/reorder: fix freeing mbuf twice
- vhost: check cmsg not null
- vhost: fix compilation issue when vhost debug enabled
- vhost: fix dead lock on closing in server mode
- vhost: fix device cleanup at stop

- vhost: fix log macro name conflict
- vhost: fix offset while mmaping log base address
- vhost: fix realloc failure
- vhost: fix typo in comment
- vhost: improve dirty pages logging performance

2.10.8 16.11.8

- app/testpmd: fix DCB config
- app/testpmd: fix VLAN TCI mask set error for FDIR
- crypto/qat: fix checks for 3GPP algo bit params
- doc: fix bonding command in testpmd
- eal: fix bitmap documentation
- eal: fix return codes on thread naming failure
- eal/linux: fix invalid syntax in interrupts
- ethdev: check queue stats mapping input arguments
- ethdev: fix a doxygen comment for port allocation
- ethdev: fix queue statistics mapping documentation
- examples/exception_path: fix out-of-bounds read
- examples/ipsec-secgw: fix bypass rule processing
- examples/ipsec-secgw: fix IPv4 checksum at Tx
- examples/l3fwd: remove useless include
- examples/multi_process: build l2fwd_fork app
- hash: fix a multi-writer race condition
- hash: fix doxygen of return values
- hash: fix key slot size accuracy
- hash: fix multiwriter lock memory allocation
- kni: fix build on RHEL 7.5
- kni: fix build with gcc 8.1
- kni: fix crash with null name
- maintainers: claim maintainership for ARM v7 and v8
- maintainers: update for Mellanox PMDs
- mbuf: fix typo in IPv6 macro comment
- mk: fix permissions when using make install
- net/bnx2x: fix FW command timeout during stop

- net/bnxt: check access denied for HWRM commands
- net/bnxt: fix close operation
- net/bnxt: fix HW Tx checksum offload check
- net/bnxt: fix incorrect IO address handling in Tx
- net/bnxt: fix RETA size
- net/bnxt: fix Rx ring count limitation
- net/bnxt: fix Tx with multiple mbuf
- net/bonding: do not clear active slave count
- net/bonding: fix MAC address reset
- net/bonding: fix race condition
- net/cxgbe/base: update flash part information
- net/cxgbe: fix init failure due to new flash parts
- net/ena: change memory type
- net/ena: check pointer before memset
- net/ena: fix GENMASK_ULL macro
- net/ena: fix SIGFPE with 0 Rx queue
- net/ena: set link speed as none
- net/enic: do not overwrite admin Tx queue limit
- net/i40e: do not reset device info data
- net/i40e: fix check of flow director programming status
- net/i40e: fix link speed
- net/i40e: fix shifts of 32-bit value
- net/i40e: revert fix of flow director check
- net/i40e: workaround performance degradation
- net/ixgbe: fix mask bits register set error for FDIR
- net/ixgbe: fix tunnel id format error for FDIR
- net/ixgbe: fix tunnel type set error for FDIR
- net/nfp: fix field initialization in Tx descriptor
- net/null: add MAC address setting fake operation
- net/pcap: fix multiple queues
- net/qede/base: fix GRC attention callback
- net/qede: fix default extended VLAN offload config
- net/qede: fix MAC address removal failure message
- net: rename u16 to fix shadowed declaration

- net/thunderx: avoid sq door bell write on zero packet
- net/thunderx: fix build with gcc optimization on
- Revert "net/i40e: fix packet count for PF"
- test/bonding: assign non-zero MAC to null devices
- test/crypto: fix device id when stopping port
- test: fix EAL flags autotest on FreeBSD
- test: fix uninitialized port configuration
- test/hash: fix multiwriter with non consecutive cores
- test/hash: fix potential memory leak
- test/virtual_pmd: add MAC address setting fake op
- vhost: fix missing increment of log cache count

2.10.9 16.11.9

- acl: forbid rule with priority zero
- app/testpmd: fix csum parse-tunnel command invocation
- app/testpmd: fix displaying RSS hash functions
- app/testpmd: fix duplicate exit
- app/testpmd: fix L4 length for UDP checksum
- app/testpmd: optimize mbuf pool allocation
- build: enable ARM NEON flag when __aarch64__ defined
- bus/pci: fix allocation of device path
- config: enable more than 128 cores for arm64
- config: make AVX and AVX512 configurable
- doc: add VFIO in ENA guide
- doc: fix formatting in IP reassembly app guide
- doc: fix NUMA library name in Linux guide
- doc: fix typo in testpmd guide
- doc: fix wrong usage of bind command
- eal: fix build
- eal: fix build with gcc 9.0
- eal: fix build with -O1
- eal: introduce rte version of fls
- eal/linux: fix memory leak of logid
- eal/linux: handle UIO read failure in interrupt handler

- eal: use correct data type for bitmap slab operations
- ethdev: fix doxygen comment to be with structure
- ethdev: fix invalid configuration after failure
- ethdev: fix queue start and stop
- examples/ipv4_multicast: fix leak of cloned packets
- examples/vhost: remove unnecessary constant
- fix dpdk.org URLs
- fix global variable issues
- hash: fix key store element alignment
- hash: remove unnecessary pause
- igb_uio: fix unexpected removal for hot-unplug
- igb_uio: issue FLR during open and release of device file
- igb_uio: remove device reset in open
- igb_uio: remove device reset in release
- ip_frag: check fragment length of incoming packet
- ip_frag: fix overflow in key comparison
- ip_frag: use key length for key comparison
- kni: fix build on CentOS 7.4
- kni: fix build on Linux < 3.14
- kni: fix build on Linux 4.19
- kni: fix build on Suse 12 SP3
- kni: fix kernel FIFO synchronization
- kni: fix possible uninitialized variable
- kni: fix SLE version detection
- kvargs: fix processing a null list
- mk: disable gcc AVX512F support
- net/bnx2x: fix call to link handling periodic function
- net/bnx2x: fix logging to include device name
- net/bnx2x: fix to add PHY lock
- net/bnx2x: fix to disable further interrupts
- net/bnx2x: fix VF link state update
- net/bnxt: fix uninitialized pointer access in Tx
- net/bnxt: reduce polling interval for valid bit
- net/bnxt: remove excess log messages

- net/bnxt: set a VNIC as default only once
- net/bnxt: set MAC filtering as outer for non tunnel frames
- net/bonding: do not ignore RSS key on device config
- net/bonding: fix crash when stopping mode 4 port
- net/bonding: fix Rx slave fairness
- net/bonding: reduce slave starvation on Rx poll
- net/bonding: stop and deactivate slaves on stop
- net/bonding: support matching QinQ ethertype
- net/bonding: use evenly distributed default RSS RETA
- net/e1000/base: fix uninitialized variable
- net/e1000: do not error out if Rx drop enable is set
- net/ena: fix passing RSS hash to mbuf
- net/enic: add dependency on librte_kvargs
- net/enic: add devarg to specify ingress VLAN rewrite mode
- net/enic: do not use non-standard integer types
- net/enic: set Rx VLAN offload flag for non-stripped packets
- net: fix build with pedantic
- net/i40e/base: correct global reset timeout calculation
- net/i40e/base: fix partition id calculation for X722
- net/i40e/base: gracefully clean the resources
- net/i40e/base: properly clean resources
- net/i40e: enable loopback function for X722 MAC
- net/i40e: fix send admin queue command before init
- net/i40e: fix X710 Rx after reading some registers
- net/i40e: keep promiscuous on if allmulticast is enabled
- net/i40e: update Tx offload mask
- net/igb: update Tx offload mask
- net/ixgbe: fix maximum wait time in comment
- net/ixgbe: fix TDH register write
- net/ixgbe: update Tx offload mask
- net/ixgbevf: fix link state
- net/ixgbe: wait longer for link after fiber MAC setup
- net/mlx5: fix build on PPC64
- net/nfp: fix live MAC changes not supported

- net/nfp: fix mbuf flags with checksum good
- net/nfp: fix RSS
- net/thunderx: fix Tx desc corruption in scatter-gather mode
- net/vhost: fix parameters string
- net/virtio: add missing supported features
- net/virtio: register/unregister intr handler on start/stop
- net/virtio-user: do not reset owner when driver resets
- net/virtio-user: fix typo in error message
- pci: fix parsing of address without function number
- test/crypto: fix number of queue pairs
- test/hash: fix bucket size in perf test
- test/hash: fix build
- test/kni: check module dependency
- test/reorder: fix out of bound access
- usertools: check for lspci dependency
- version: 16.11.9-rc1
- version: 16.11.9-rc2
- vfio: fix build
- vfio: fix build on old kernel
- vhost: fix corner case for enqueue operation
- vhost: fix payload size of reply
- vhost: remove unneeded null pointer check
- vhost-user: drop connection on message handling failures
- vhost-user: fix false negative in handling user messages

2.11 16.11.10 Release Notes

2.11.1 16.11.10 Fixes

- vhost: validate virtqueue size
- vhost: add number of fds to vhost-user messages
- vhost: fix possible denial of service by leaking FDs - CVE-2019-14818
- vhost: fix possible denial of service on SET_VRING_NUM - CVE-2019-14818

2.11.2 16.11.10 Validation

- Tested with two testpmd instances, one with Vhost PMD, the other with Virtio-user PMD. Intialization goes well, and packets flow.

2.12 16.11.11 Release Notes

2.12.1 16.11.11 Fixes

- vhost: fix vring requests validation broken if no FD

2.12.2 16.11.11 Validation

- virtio/vhost regression tests by Intel: * http://doc.dpdk.org/dts/test_plans/virtio_pvp_regression_test_plan.html * http://doc.dpdk.org/dts/test_plans/vhost_dequeue_zero_copy_test_plan.html * http://doc.dpdk.org/dts/test_plans/vm2vm_virtio_pmd_test_plan.html

CHAPTER

THREE

DPDK RELEASE 16.07

3.1 New Features

- **Removed the mempool cache memory if caching is not being used.**

  The size of the mempool structure is reduced if the per-lcore cache is disabled.

- **Added mempool external cache for non-EAL thread.**

  Added new functions to create, free or flush a user-owned mempool cache for non-EAL threads. Previously the caching was always disabled on these threads.

- **Changed the memory allocation scheme in the mempool library.**

  - Added the ability to allocate a large mempool in fragmented virtual memory.
  - Added new APIs to populate a mempool with memory.
  - Added an API to free a mempool.
  - Modified the API of the `rte_mempool_obj_iter()` function.
  - Dropped the specific Xen Dom0 code.
  - Dropped the specific anonymous mempool code in testpmd.

- **Added a new driver for Broadcom NetXtreme-C devices.**

  Added the new bnxt driver for Broadcom NetXtreme-C devices. See the "Network Interface Controller Drivers" document for more details on this new driver.

- **Added a new driver for ThunderX nicvf devices.**

  Added the new thunderx net driver for ThunderX nicvf devices. See the "Network Interface Controller Drivers" document for more details on this new driver.

- **Added mailbox interrupt support for ixgbe and igb VFs.**

  When the physical NIC link comes up or down, the PF driver will send a mailbox message to notify each VF. To handle this link up/down event, support have been added for a mailbox interrupt to receive the message and allow the application to register a callback for it.

- **Updated the ixgbe base driver.**

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

  - Added sgmii link for X550.
  - Added MAC link setup for X550a SFP and SFP+.

  - Added KR support for X550em_a.
  - Added new PHY definitions for M88E1500.
  - Added support for the VLVF to be bypassed when adding/removing a VFTA entry.
  - Added X550a flow control auto negotiation support.

- **Updated the i40e base driver.**

  Updated the i40e base driver including support for new devices IDs.

- **Updated the enic driver.**

  The enic driver was updated with changes including the following:

  - Optimized the Tx function.
  - Added Scattered Rx capability.
  - Improved packet type identification.
  - Added MTU update in non Scattered Rx mode and enabled MTU of up to 9208 with UCS Software release 2.2 on 1300 series VICs.

- **Updated the mlx5 driver.**

  The mlx5 driver was updated with changes including the following:

  - Data path was refactored to bypass Verbs to improve RX and TX performance.
  - Removed compilation parameters for inline send, `MLX5_MAX_INLINE`, and added command line parameter instead, `txq_inline`.
  - Improved TX scatter gather support: Removed compilation parameter `MLX5_PMD_SGE_WR_N`. Scatter-gather elements is set to the maximum value the NIC supports. Removed linearization logic, this decreases the memory consumption of the PMD.
  - Improved jumbo frames support, by dynamically setting RX scatter gather elements according to the MTU and mbuf size, no need for compilation parameter `MLX5_PMD_SGE_WR_N`

- **Added support for virtio on IBM POWER8.**

  The ioports are mapped in memory when using Linux UIO.

- **Added support for Virtio in containers.**

  Add a new virtual device, named virtio_user, to support virtio for containers.

  Known limitations:

  - Control queue and multi-queue are not supported yet.
  - Doesn't work with `--huge-unlink`.
  - Doesn't work with `--no-huge`.
  - Doesn't work when there are more than `VHOST_MEMORY_MAX_NREGIONS(8)` hugepages.
  - Root privilege is required for sorting hugepages by physical address.
  - Can only be used with the vhost user backend.

- **Added vhost-user client mode.**

  DPDK vhost-user now supports client mode as well as server mode. Client mode is enabled when the `RTE_VHOST_USER_CLIENT` flag is set while calling `rte_vhost_driver_register`.

  When DPDK vhost-user restarts from an normal or abnormal exit (such as a crash), the client mode allows DPDK to establish the connection again. Note that QEMU version v2.7 or above is required for this feature.

  DPDK vhost-user will also try to reconnect by default when:

  - The first connect fails (for example when QEMU is not started yet).
  - The connection is broken (for example when QEMU restarts).

  It can be turned off by setting the `RTE_VHOST_USER_NO_RECONNECT` flag.

- **Added NSH packet recognition in i40e.**

- **Added AES-CTR support to AESNI MB PMD.**

  Now AESNI MB PMD supports 128/192/256-bit counter mode AES encryption and decryption.

- **Added AES counter mode support for Intel QuickAssist devices.**

  Enabled support for the AES CTR algorithm for Intel QuickAssist devices. Provided support for algorithm-chaining operations.

- **Added KASUMI SW PMD.**

  A new Crypto PMD has been added, which provides KASUMI F8 (UEA1) ciphering and KASUMI F9 (UIA1) hashing.

- **Added multi-writer support for RTE Hash with Intel TSX.**

  The following features/modifications have been added to rte_hash library:

  - Enabled application developers to use an extra flag for `rte_hash` creation to specify default behavior (multi-thread safe/unsafe) with the `rte_hash_add_key` function.
  - Changed the Cuckoo Hash Search algorithm to breadth first search for multi-writer routines and split Cuckoo Hash Search and Move operations in order to reduce transactional code region and improve TSX performance.
  - Added a hash multi-writer test case to the test app.

- **Improved IP Pipeline Application.**

  The following features have been added to the ip_pipeline application:

  - Configure the MAC address in the routing pipeline and automatic route updates with change in link state.
  - Enable RSS per network interface through the configuration file.
  - Streamline the CLI code.

- **Added keepalive enhancements.**

  Added support for reporting of core states other than “dead” to monitoring applications, enabling the support of broader liveness reporting to external processes.

- **Added packet capture framework.**
  - A new library `librte_pdump` is added to provide a packet capture API.
  - A new `app/pdump` tool is added to demonstrate capture packets in DPDK.
- **Added floating VEB support for i40e PF driver.**

  A “floating VEB” is a special Virtual Ethernet Bridge (VEB) which does not have an upload port, but instead is used for switching traffic between virtual functions (VFs) on a port.

  For information on this feature, please see the “I40E Poll Mode Driver” section of the “Network Interface Controller Drivers” document.
- **Added support for live migration of a VM with SRIOV VF.**

  Live migration of a VM with Virtio and VF PMD’s using the bonding PMD.

3.2 Resolved Issues

3.2.1 EAL

- **igb_uio: Fixed possible mmap failure for Linux >= 4.5.**

  The mmaping of the iomem range of the PCI device fails for kernels that enabled the `CONFIG_IO_STRICT_DEVMEM` option. The error seen by the user is as similar to the following:

  ```
  EAL: pci_map_resource():

      cannot mmap(39, 0x7f1c51800000, 0x100000, 0x0):
      Invalid argument (0xffffffffffffffff)
  ```

  The `CONFIG_IO_STRICT_DEVMEM` kernel option was introduced in Linux v4.5.

  The issues was resolve by updating `igb_uio` to stop reserving PCI memory resources. From the kernel point of view the iomem region looks like idle and mmap works again. This matches the `uio_pci_generic` usage.

3.2.2 Drivers

- **i40e: Fixed vlan stripping from inner header.**

  Previously, for tunnel packets, such as VXLAN/NVGRE, the vlan tags of the inner header will be stripped without putting vlan info to descriptor. Now this issue is fixed by disabling vlan stripping from inner header.
- **i40e: Fixed the type issue of a single VLAN type.**

  Currently, if a single VLAN header is added in a packet, it’s treated as inner VLAN. But generally, a single VLAN header is treated as the outer VLAN header. This issue is fixed by changing corresponding register for single VLAN.
- **enic: Fixed several issues when stopping then restarting ports and queues.**

  Fixed several crashes related to stopping then restarting ports and queues. Fixed possible crash when re-configuring the number of Rx queue descriptors.

- **enic: Fixed Rx data mis-alignment if mbuf data offset modified.**

  Fixed possible Rx corruption when mbufs were returned to a pool with data offset other than RTE_PKTMBUF_HEADROOM.

- **enic: Fixed Tx IP/UDP/TCP checksum offload and VLAN insertion.**
- **enic: Fixed Rx error and missed counters.**

3.2.3 Libraries

- **mbuf: Fixed refcnt update when detaching.**

  Fix the `rte_pktmbuf_detach()` function to decrement the direct mbuf's reference counter. The previous behavior was not to affect the reference counter. This lead to a memory leak of the direct mbuf.

3.2.4 Examples

3.2.5 Other

3.3 Known Issues

3.4 API Changes

- The following counters are removed from the `rte_eth_stats` structure:
  - `ibadcrc`
  - `ibadlen`
  - `imcasts`
  - `fdirmatch`
  - `fdirmiss`
  - `tx_pause_xon`
  - `rx_pause_xon`
  - `tx_pause_xoff`
  - `rx_pause_xoff`
- The extended statistics are fetched by ids with `rte_eth_xstats_get` after a lookup by name `rte_eth_xstats_get_names`.
- The function `rte_eth_dev_info_get` fill the new fields `nb_rx_queues` and `nb_tx_queues` in the structure `rte_eth_dev_info`.
- The vhost function `rte_vring_available_entries` is renamed to `rte_vhost_avail_entries`.
- All existing vhost APIs and callbacks with `virtio_net` struct pointer as the parameter have been changed due to the ABI refactoring described below. It is replaced by `int vid`.

- The function `rte_vhost_enqueue_burst` no longer supports concurrent enqueuing packets to the same queue.
- The function `rte_eth_dev_set_mtu` adds a new return value `-EBUSY`, which indicates the operation is forbidden because the port is running.
- The script `dpdk_nic_bind.py` is renamed to `dpdk-devbind.py`. And the script `setup.sh` is renamed to `dpdk-setup.sh`.

3.5 ABI Changes

- The `rte_port_source_params` structure has new fields to support PCAP files. It was already in release 16.04 with `RTE_NEXT_ABI` flag.
- The `rte_eth_dev_info` structure has new fields `nb_rx_queues` and `nb_tx_queues` to support the number of queues configured by software.
- A Vhost ABI refactoring has been made: the `virtio_net` structure is no longer exported directly to the application. Instead, a handle, `vid`, has been used to represent this structure internally.

3.6 Shared Library Versions

The libraries prepended with a plus sign were incremented in this version.

```
+ libethdev.so.4
  librte_acl.so.2
  librte_cfgfile.so.2
  librte_cmdline.so.2
  librte_cryptodev.so.1
  librte_distributor.so.1
  librte_eal.so.2
  librte_hash.so.2
  librte_ip_frag.so.1
  librte_ivshmem.so.1
  librte_jobstats.so.1
  librte_kni.so.2
  librte_kvargs.so.1
  librte_lpm.so.2
  librte_mbuf.so.2
+ librte_mempool.so.2
  librte_meter.so.1
  librte_pdump.so.1
  librte_pipeline.so.3
  librte_pmd_bond.so.1
  librte_pmd_ring.so.2
+ librte_port.so.3
  librte_power.so.1
  librte_reorder.so.1
  librte_ring.so.1
  librte_sched.so.1
  librte_table.so.2
  librte_timer.so.1
+ librte_vhost.so.3
```

3.7 Tested Platforms

1. SuperMicro 1U
   - BIOS: 1.0c
   - Processor: Intel(R) Atom(TM) CPU C2758 @ 2.40GHz
2. SuperMicro 1U
   - BIOS: 1.0a
   - Processor: Intel(R) Xeon(R) CPU D-1540 @ 2.00GHz
   - Onboard NIC: Intel(R) X552/X557-AT (2x10G)
     - Firmware-version: 0x800001cf
     - Device ID (PF/VF): 8086:15ad /8086:15a8
   - kernel driver version: 4.2.5 (ixgbe)
3. SuperMicro 2U
   - BIOS: 1.0a
   - Processor: Intel(R) Xeon(R) CPU E5-4667 v3 @ 2.00GHz
4. Intel(R) Server board S2600GZ
   - BIOS: SE5C600.86B.02.02.0002.122320131210
   - Processor: Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz
5. Intel(R) Server board W2600CR
   - BIOS: SE5C600.86B.02.01.0002.082220131453
   - Processor: Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz
6. Intel(R) Server board S2600CWT
   - BIOS: SE5C610.86B.01.01.0009.060120151350
   - Processor: Intel(R) Xeon(R) CPU E5-2699 v3 @ 2.30GHz
7. Intel(R) Server board S2600WTT
   - BIOS: SE5C610.86B.01.01.0005.101720141054
   - Processor: Intel(R) Xeon(R) CPU E5-2699 v3 @ 2.30GHz
8. Intel(R) Server board S2600WTT
   - BIOS: SE5C610.86B.11.01.0044.090120151156
   - Processor: Intel(R) Xeon(R) CPU E5-2695 v4 @ 2.10GHz

3.8 Tested NICs

1. Intel(R) Ethernet Controller X540-AT2
   - Firmware version: 0x80000389

   - Device id (pf): 8086:1528
   - Driver version: 3.23.2 (ixgbe)

2. Intel(R) 82599ES 10 Gigabit Ethernet Controller
   - Firmware version: 0x61bf0001
   - Device id (pf/vf): 8086:10fb / 8086:10ed
   - Driver version: 4.0.1-k (ixgbe)
3. Intel(R) Corporation Ethernet Connection X552/X557-AT 10GBASE-T
   - Firmware version: 0x800001cf
   - Device id (pf/vf): 8086:15ad / 8086:15a8
   - Driver version: 4.2.5 (ixgbe)
4. Intel(R) Ethernet Converged Network Adapter X710-DA4 (4x10G)
   - Firmware version: 5.04
   - Device id (pf/vf): 8086:1572 / 8086:154c
   - Driver version: 1.4.26 (i40e)
5. Intel(R) Ethernet Converged Network Adapter X710-DA2 (2x10G)
   - Firmware version: 5.04
   - Device id (pf/vf): 8086:1572 / 8086:154c
   - Driver version: 1.4.25 (i40e)
6. Intel(R) Ethernet Converged Network Adapter XL710-QDA1 (1x40G)
   - Firmware version: 5.04
   - Device id (pf/vf): 8086:1584 / 8086:154c
   - Driver version: 1.4.25 (i40e)
7. Intel(R) Ethernet Converged Network Adapter XL710-QDA2 (2X40G)
   - Firmware version: 5.04
   - Device id (pf/vf): 8086:1583 / 8086:154c
   - Driver version: 1.4.25 (i40e)
8. Intel(R) Corporation I350 Gigabit Network Connection
   - Firmware version: 1.48, 0x800006e7
   - Device id (pf/vf): 8086:1521 / 8086:1520
   - Driver version: 5.2.13-k (igb)
9. Intel(R) Ethernet Multi-host Controller FM10000
   - Firmware version: N/A
   - Device id (pf/vf): 8086:15d0
   - Driver version: 0.17.0.9 (fm10k)

3.9 Tested OSes

- CentOS 7.0
- Fedora 23
- Fedora 24
- FreeBSD 10.3
- Red Hat Enterprise Linux 7.2
- SUSE Enterprise Linux 12
- Ubuntu 15.10
- Ubuntu 16.04 LTS
- Wind River Linux 8

CHAPTER

FOUR

DPDK RELEASE 16.04

4.1 New Features

- **Added function to check primary process state.**

  A new function `rte_eal_primary_proc_alive()` has been added to allow the user to detect if a primary process is running. Use cases for this feature include fault detection, and monitoring using secondary processes.

- **Enabled bulk allocation of mbufs.**

  A new function `rte_pktmbuf_alloc_bulk()` has been added to allow the user to bulk allocate mbufs.

- **Added device link speed capabilities.**

  The structure `rte_eth_dev_info` now has a `speed_capa` bitmap, which allows the application to determine the supported speeds of each device.

- **Added bitmap of link speeds to advertise.**

  Added a feature to allow the definition of a set of advertised speeds for auto-negotiation, explicitly disabling link auto-negotiation (single speed) and full auto-negotiation.

- **Added new poll-mode driver for Amazon Elastic Network Adapters (ENA).**

  The driver operates for a variety of ENA adapters through feature negotiation with the adapter and upgradable commands set. The ENA driver handles PCI Physical and Virtual ENA functions.

- **Restored vmxnet3 TX data ring.**

  TX data ring has been shown to improve small packet forwarding performance on the vSphere environment.

- **Added vmxnet3 TX L4 checksum offload.**

  Added support for TCP/UDP checksum offload to vmxnet3.

- **Added vmxnet3 TSO support.**

  Added support for TSO to vmxnet3.

- **Added vmxnet3 support for jumbo frames.**

  Added support for linking multi-segment buffers together to handle Jumbo packets.

- **Enabled Virtio 1.0 support.**

  Enabled Virtio 1.0 support for Virtio pmd driver.

- **Supported Virtio for ARM.**

  Enabled Virtio support for ARMv7/v8. Tested for ARM64. Virtio for ARM supports VFIO-noiommu mode only. Virtio can work with other non-x86 architectures as well, like PowerPC.

- **Supported Virtio offload in vhost-user.**

  Added the offload and negotiation of checksum and TSO between vhost-user and vanilla Linux Virtio guest.

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

- **Added vhost driver.**

  Added a virtual PMD that wraps `librte_vhost`.

- **Added multicast promiscuous mode support on VF for ixgbe.**

  Added multicast promiscuous mode support for the ixgbe VF driver so all VFs can receive the multicast packets.

  Please note if you want to use this promiscuous mode, you need both PF and VF driver to support it. The reason is that this VF feature is configured in the PF. If you use kernel PF driver and the dpdk VF driver, make sure the kernel PF driver supports VF multicast promiscuous mode. If you use dpdk PF and dpdk VF ensure the PF driver is the same version as the VF.

- **Added support for E-tag on X550.**

  E-tag is defined in 802.1BR - Bridge Port Extension.

  This feature is for the VF, but the settings are on the PF. It means the CLIs should be used on the PF, but some of their effects will be shown on the VF. The forwarding of E-tag packets based on GRP and E-CID_base will have an effect on the PF. Theoretically, the E-tag packets can be forwarded to any pool/queue but normally we'd like to forward the packets to the pools/queues belonging to the VFs. And E-tag insertion and stripping will have an effect on VFs. When a VF receives E-tag packets it should strip the E-tag. When the VF transmits packets, it should insert the E-tag. Both actions can be offloaded.

  When we want to use this E-tag support feature, the forwarding should be enabled to forward the packets received by the PF to the indicated VFs. And insertion and stripping should be enabled for VFs to offload the effort to hardware.

  Features added:

  - Support E-tag offloading of insertion and stripping.
  - Support Forwarding E-tag packets to pools based on GRP and E-CID_base.

- **Added support for VxLAN and NVGRE checksum off-load on X550.**

  - Added support for VxLAN and NVGRE RX/TX checksum off-load on X550. RX/TX checksum off-load is provided on both inner and outer IP header and TCP header.
  - Added functions to support VxLAN port configuration. The default VxLAN port number is 4789 but this can be updated programmatically.

- **Added support for new X550EM_a devices.**

  Added support for new X550EM_a devices and their MAC types, X550EM_a and X550EM_a_vf. Updated the relevant PMD to use the new devices and MAC types.

- **Added x550em_x V2 device support.**

  Added support for x550em_x V2 device. Only x550em_x V1 was supported before. A mask for V1 and V2 is defined and used to support both.

- **Supported link speed auto-negotiation on X550EM_X**

  Normally the auto-negotiation is supported by firmware and software doesn't care about it. But on x550em_x, firmware doesn't support auto-negotiation. As the ports of x550em_x are 10GbE, if we connect the port with a peer which is 1GbE, the link will always be down. We added the support for auto-negotiation by software to avoid this link down issue.

- **Added software-firmware sync on X550EM_a.**

  Added support for software-firmware sync for resource sharing. Use the PHY token, shared between software-firmware for PHY access on X550EM_a.

- **Updated the i40e base driver.**

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

  - Use RX control AQ commands to read/write RX control registers.
  - Add new X722 device IDs, and removed X710 one was never used.
  - Expose registers for HASH/FD input set configuring.

- **Enabled PCI extended tag for i40e.**

  Enabled extended tag for i40e by checking and writing corresponding PCI config space bytes, to boost the performance. The legacy method of reading/writing sysfile supported by kernel module igb_uio is now deprecated.

- **Added i40e support for setting mac addresses.**
- **Added dump of i40e registers and EEPROM.**
- **Supported ether type setting of single and double VLAN for i40e**
- **Added VMDQ DCB mode in i40e.**

  Added support for DCB in VMDQ mode to i40e driver.

- **Added i40e VEB switching support.**
- **Added Flow director enhancements in i40e.**
- **Added PF reset event reporting in i40e VF driver.**
- **Added fm10k RX interrupt support.**
- **Optimized fm10k TX.**

  Optimized fm10k TX by freeing multiple mbufs at a time.

- **Handled error flags in fm10k vector RX.**

  Parse error flags in RX descriptor and set error bits in mbuf with vector instructions.

- **Added fm10k FTAG based forwarding support.**
- **Added mlx5 flow director support.**

  Added flow director support (RTE_FDIR_MODE_PERFECT and RTE_FDIR_MODE_PERFECT_MAC_VLAN).

  Only available with Mellanox OFED >= 3.2.
- **Added mlx5 RX VLAN stripping support.**

  Added support for RX VLAN stripping.

  Only available with Mellanox OFED >= 3.2.
- **Added mlx5 link up/down callbacks.**

  Implemented callbacks to bring link up and down.
- **Added mlx5 support for operation in secondary processes.**

  Implemented TX support in secondary processes (like mlx4).
- **Added mlx5 RX CRC stripping configuration.**

  Until now, CRC was always stripped. It can now be configured.

  Only available with Mellanox OFED >= 3.2.
- **Added mlx5 optional packet padding by HW.**

  Added an option to make PCI bus transactions rounded to a multiple of a cache line size for better alignment.

  Only available with Mellanox OFED >= 3.2.
- **Added mlx5 TX VLAN insertion support.**

  Added support for TX VLAN insertion.

  Only available with Mellanox OFED >= 3.2.
- **Changed szedata2 driver type from vdev to pdev.**

  Previously szedata2 device had to be added by `--vdev` option. Now szedata2 PMD recognizes the device automatically during EAL initialization.
- **Added szedata2 functions for setting link up/down.**
- **Added szedata2 promiscuous and allmulticast modes.**
- **Added af_packet dynamic removal function.**

  An af_packet device can now be detached using the API, like other PMD devices.
- **Increased number of next hops for LPM IPv4 to 2^24.**

  The `next_hop` field has been extended from 8 bits to 24 bits for IPv4.
- **Added support of SNOW 3G (UEA2 and UIA2) for Intel Quick Assist devices.**

  Enabled support for the SNOW 3G wireless algorithm for Intel Quick Assist devices. Support for cipher-only and hash-only is also provided along with algorithm-chaining operations.

- **Added SNOW3G SW PMD.**

  A new Crypto PMD has been added, which provides SNOW 3G UEA2 ciphering and SNOW3G UIA2 hashing.

- **Added AES GCM PMD.**

  Added new Crypto PMD to support AES-GCM authenticated encryption and authenticated decryption in software.

- **Added NULL Crypto PMD**

  Added new Crypto PMD to support null crypto operations in software.

- **Improved IP Pipeline Application.**

  The following features have been added to ip_pipeline application;

  - Added CPU utilization measurement and idle cycle rate computation.
  - Added link identification support through existing port-mask option or by specifying PCI device in every LINK section in the configuration file.
  - Added load balancing support in passthrough pipeline.

- **Added IPsec security gateway example.**

  Added a new application implementing an IPsec Security Gateway.

4.2 Resolved Issues

4.2.1 Drivers

- **ethdev: Fixed overflow for 100Gbps.**

  100Gbps in Mbps (100000) was exceeding the 16-bit max value of `link_speed` in `rte_eth_link`.

- **ethdev: Fixed byte order consistency between fdir flow and mask.**

  Fixed issue in ethdev library where the structure for setting fdir's mask and flow entry was not consistent in byte ordering.

- **cxgbe: Fixed crash due to incorrect size allocated for RSS table.**

  Fixed a segfault that occurs when accessing part of port 0's RSS table that gets overwritten by subsequent port 1's part of the RSS table due to incorrect size allocated for each entry in the table.

- **cxgbe: Fixed setting wrong device MTU.**

  Fixed an incorrect device MTU being set due to the Ethernet header and CRC lengths being added twice.

- **ixgbe: Fixed zeroed VF mac address.**

  Resolved an issue where the VF MAC address is zeroed out in cases where the VF driver is loaded while the PF interface is down. The solution is to only set it when we get an ACK from the PF.

- **ixgbe: Fixed setting flow director flag twice.**

  Resolved an issue where packets were being dropped when switching to perfect filters mode.

- **ixgbe: Set MDIO speed after MAC reset.**

  The MDIO clock speed must be reconfigured after the MAC reset. The MDIO clock speed becomes invalid, therefore the driver reads invalid PHY register values. The driver now set the MDIO clock speed prior to initializing PHY ops and again after the MAC reset.

- **ixgbe: Fixed maximum number of available TX queues.**

  In IXGBE, the maximum number of TX queues varies depending on the NIC operating mode. This was not being updated in the device information, providing an incorrect number in some cases.

- **i40e: Generated MAC address for each VFs.**

  It generates a MAC address for each VFs during PF host initialization, and keeps the VF MAC address the same among different VF launch.

- **i40e: Fixed failure of reading/writing RX control registers.**

  Fixed i40e issue of failing to read/write rx control registers when under stress with traffic, which might result in application launch failure.

- **i40e: Enabled vector driver by default.**

  Previously, vector driver was disabled by default as it couldn't fill packet type info for l3fwd to work well. Now there is an option for l3fwd to analyze the packet type so the vector driver is enabled by default.

- **i40e: Fixed link info of VF.**

  Previously, the VF's link speed stayed at 10GbE and status always was up. It did not change even when the physical link's status changed. Now this issue is fixed to make VF's link info consistent with physical link.

- **mlx5: Fixed possible crash during initialization.**

  A crash could occur when failing to allocate private device context.

- **mlx5: Added port type check.**

  Added port type check to prevent port initialization on non-Ethernet link layers and to report an error.

- **mlx5: Applied VLAN filtering to broadcast and IPv6 multicast flows.**

  Prevented reception of multicast frames outside of configured VLANs.

- **mlx5: Fixed RX checksum offload in non L3/L4 packets.**

  Fixed report of bad checksum for packets of unknown type.

- **aesni_mb: Fixed wrong return value when creating a device.**

  The `cryptodev_aesni_mb_init()` function was returning the device id of the device created, instead of 0 (on success) that `rte_eal_vdev_init()` expects. This made it impossible to create more than one aesni_mb device from the command line.

- **qat: Fixed AES GCM decryption.**

  Allowed AES GCM on the cryptodev API, but in some cases gave invalid results due to incorrect IV setting.

4.2.2 Libraries

- **hash: Fixed CRC32c hash computation for non multiple of 4 bytes sizes.**

  Fix crc32c hash functions to return a valid crc32c value for data lengths not a multiple of 4 bytes.

- **hash: Fixed hash library to support multi-process mode.**

  Fix hash library to support multi-process mode, using a jump table, instead of storing a function pointer to the key compare function. Multi-process mode only works with the built-in compare functions, however a custom compare function (not in the jump table) can only be used in single-process mode.

- **hash: Fixed return value when allocating an existing hash table.**

  Changed the `rte_hash*_create()` functions to return `NULL` and set `rte_errno` to `EEXIST` when the object name already exists. This is the behavior described in the API documentation in the header file. The previous behavior was to return a pointer to the existing object in that case, preventing the caller from knowing if the object had to be freed or not.

- **lpm: Fixed return value when allocating an existing object.**

  Changed the `rte_lpm*_create()` functions to return `NULL` and set `rte_errno` to `EEXIST` when the object name already exists. This is the behavior described in the API documentation in the header file. The previous behavior was to return a pointer to the existing object in that case, preventing the caller from knowing if the object had to be freed or not.

- **librte_port: Fixed segmentation fault for ring and ethdev writer nodrop.**

  Fixed core dump issue on txq and swq when dropless is set to yes.

4.2.3 Examples

- **l3fwd-power: Fixed memory leak for non-IP packet.**

  Fixed issue in l3fwd-power where, on receiving packets of types other than IPv4 or IPv6, the mbuf was not released, and caused a memory leak.

- **l3fwd: Fixed using packet type blindly.**

  l3fwd makes use of packet type information without querying if devices or PMDs really set it. For those devices that don't set ptypes, add an option to parse it.

- **examples/vhost: Fixed frequent mbuf allocation failure.**

  The vhost-switch often fails to allocate mbuf when dequeue from vring because it wrongly calculates the number of mbufs needed.

4.3 API Changes

- The ethdev statistics counter `imissed` is considered to be independent of `ierrors`. All drivers are now counting the missed packets only once, i.e. drivers will not increment ierrors anymore for missed packets.
- The ethdev structure `rte_eth_dev_info` was changed to support device speed capabilities.
- The ethdev structures `rte_eth_link` and `rte_eth_conf` were changed to support the new link API.
- The functions `rte_eth_dev_udp_tunnel_add` and `rte_eth_dev_udp_tunnel_delete` have been renamed into `rte_eth_dev_udp_tunnel_port_add` and `rte_eth_dev_udp_tunnel_port_delete`.
- The `outer_mac` and `inner_mac` fields in structure `rte_eth_tunnel_filter_conf` are changed from pointer to struct in order to keep code's readability.
- The fields in ethdev structure `rte_eth_fdir_masks` were changed to be in big endian.
- A parameter `vlan_type` has been added to the function `rte_eth_dev_set_vlan_ether_type`.
- The af_packet device init function is no longer public. The device should be attached via the API.
- The LPM `next_hop` field is extended from 8 bits to 24 bits for IPv4 while keeping ABI compatibility.
- A new `rte_lpm_config` structure is used so the LPM library will allocate exactly the amount of memory which is necessary to hold application's rules. The previous ABI is kept for compatibility.
- The prototype for the pipeline input port, output port and table action handlers are updated: the pipeline parameter is added, the packets mask parameter has been either removed or made input-only.

4.4 ABI Changes

- The RETA entry size in `rte_eth_rss_reta_entry64` has been increased from 8-bit to 16-bit.
- The ethdev flow director structure `rte_eth_fdir_flow` structure was changed. New fields were added to extend flow director's input set.
- The cmdline buffer size has been increase from 256 to 512.

4.5 Shared Library Versions

The libraries prepended with a plus sign were incremented in this version.

```
+ libethdev.so.3
  librte_acl.so.2
  librte_cfgfile.so.2
+ librte_cmdline.so.2
  librte_distributor.so.1
  librte_eal.so.2
  librte_hash.so.2
  librte_ip_frag.so.1
  librte_ivshmem.so.1
  librte_jobstats.so.1
  librte_kni.so.2
  librte_kvargs.so.1
  librte_lpm.so.2
  librte_mbuf.so.2
  librte_mempool.so.1
  librte_meter.so.1
+ librte_pipeline.so.3
  librte_pmd_bond.so.1
  librte_pmd_ring.so.2
  librte_port.so.2
  librte_power.so.1
  librte_reorder.so.1
  librte_ring.so.1
  librte_sched.so.1
  librte_table.so.2
  librte_timer.so.1
  librte_vhost.so.2
```

4.6 Tested Platforms

1. SuperMicro 1U
   - BIOS: 1.0c
   - Processor: Intel(R) Atom(TM) CPU C2758 @ 2.40GHz
2. SuperMicro 1U
   - BIOS: 1.0a
   - Processor: Intel(R) Xeon(R) CPU D-1540 @ 2.00GHz
   - Onboard NIC: Intel(R) X552/X557-AT (2x10G)
     - Firmware-version: 0x800001cf
     - Device ID (PF/VF): 8086:15ad /8086:15a8
   - kernel driver version: 4.2.5 (ixgbe)
3. SuperMicro 1U
   - BIOS: 1.0a
   - Processor: Intel(R) Xeon(R) CPU E5-4667 v3 @ 2.00GHz
4. Intel(R) Server board S2600GZ
   - BIOS: SE5C600.86B.02.02.0002.122320131210
   - Processor: Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz
5. Intel(R) Server board W2600CR

   - BIOS: SE5C600.86B.02.01.0002.082220131453
   - Processor: Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz

6. Intel(R) Server board S2600CWT
   - BIOS: SE5C610.86B.01.01.0009.060120151350
   - Processor: Intel(R) Xeon(R) CPU E5-2699 v3 @ 2.30GHz
7. Intel(R) Server board S2600WTT
   - BIOS: SE5C610.86B.01.01.0005.101720141054
   - Processor: Intel(R) Xeon(R) CPU E5-2699 v3 @ 2.30GHz
8. Intel(R) Server board S2600WTT
   - BIOS: SE5C610.86B.11.01.0044.090120151156
   - Processor: Intel(R) Xeon(R) CPU E5-2695 v4 @ 2.10GHz

4.7 Tested NICs

1. Intel(R) Ethernet Controller X540-AT2
   - Firmware version: 0x80000389
   - Device id (pf): 8086:1528
   - Driver version: 3.23.2 (ixgbe)
2. Intel(R) 82599ES 10 Gigabit Ethernet Controller
   - Firmware version: 0x61bf0001
   - Device id (pf/vf): 8086:10fb / 8086:10ed
   - Driver version: 4.0.1-k (ixgbe)
3. Intel(R) Corporation Ethernet Connection X552/X557-AT 10GBASE-T
   - Firmware version: 0x800001cf
   - Device id (pf/vf): 8086:15ad / 8086:15a8
   - Driver version: 4.2.5 (ixgbe)
4. Intel(R) Ethernet Converged Network Adapter X710-DA4 (4x10G)
   - Firmware version: 5.02 0x80002284
   - Device id (pf/vf): 8086:1572 / 8086:154c
   - Driver version: 1.4.26 (i40e)
5. Intel(R) Ethernet Converged Network Adapter X710-DA2 (2x10G)
   - Firmware version: 5.02 0x80002282
   - Device id (pf/vf): 8086:1572 / 8086:154c
   - Driver version: 1.4.25 (i40e)
6. Intel(R) Ethernet Converged Network Adapter XL710-QDA1 (1x40G)

   - Firmware version: 5.02 0x80002281
   - Device id (pf/vf): 8086:1584 / 8086:154c
   - Driver version: 1.4.25 (i40e)

7. Intel(R) Ethernet Converged Network Adapter XL710-QDA2 (2X40G)
   - Firmware version: 5.02 0x80002285
   - Device id (pf/vf): 8086:1583 / 8086:154c
   - Driver version: 1.4.25 (i40e)
8. Intel(R) 82576EB Gigabit Ethernet Controller
   - Firmware version: 1.2.1
   - Device id (pf): 8086:1526
   - Driver version: 5.2.13-k (igb)
9. Intel(R) Ethernet Controller I210
   - Firmware version: 3.16, 0x80000500, 1.304.0
   - Device id (pf): 8086:1533
   - Driver version: 5.2.13-k (igb)
10. Intel(R) Corporation I350 Gigabit Network Connection
    - Firmware version: 1.48, 0x800006e7
    - Device id (pf/vf): 8086:1521 / 8086:1520
    - Driver version: 5.2.13-k (igb)
11. Intel(R) Ethernet Multi-host Controller FM10000
    - Firmware version: N/A
    - Device id (pf/vf): 8086:15d0
    - Driver version: 0.17.0.9 (fm10k)

CHAPTER

FIVE

DPDK RELEASE 2.2

5.1 New Features

- **Introduce ARMv7 and ARMv8 architectures.**
  - It is now possible to build DPDK for the ARMv7 and ARMv8 platforms.
  - ARMv7 can be tested with virtual PMD drivers.
  - ARMv8 can be tested with virtual and physical PMD drivers.
- **Enabled freeing of ring.**

  A new function `rte_ring_free()` has been added to allow the user to free a ring if it was created with `rte_ring_create()`.
- **Added keepalive support to EAL and example application.**
- **Added experimental cryptodev API**

  The cryptographic processing of packets is provided as a preview with two drivers for:
  - Intel QuickAssist devices
  - Intel AES-NI multi-buffer library

  Due to its experimental state, the API may change without prior notice.
- **Added ethdev APIs for additional IEEE1588 support.**

  Added functions to read, write and adjust system time in the NIC. Added client slave sample application to demonstrate the IEEE1588 functionality.
- **Extended Statistics.**

  Defined an extended statistics naming scheme to store metadata in the name string of each statistic. Refer to the Extended Statistics section of the Programmers Guide for more details.

  Implemented the extended statistics API for the following PMDs:
  - `igb`
  - `igbvf`
  - `i40e`
  - `i40evf`
  - `fm10k`

  - `virtio`

- **Added API in ethdev to retrieve RX/TX queue information.**
  - Added the ability for the upper layer to query RX/TX queue information.
  - Added new fields in `rte_eth_dev_info` to represent information about RX/TX descriptors min/max/align numbers, per queue, for the device.
- **Added RSS dynamic configuration to bonding.**
- **Updated the e1000 base driver.**

  The e1000 base driver was updated with several features including the following:
  - Added new i218 devices
  - Allowed both ULP and EEE in Sx state
  - Initialized 88E1543 (Marvell 1543) PHY
  - Added flags to set EEE advertisement modes
  - Supported inverted format ETrackId
  - Added bit to disable packetbuffer read
  - Added defaults for i210 RX/TX PBSIZE
  - Check more errors for ESB2 init and reset
  - Check more NVM read errors
  - Return code after setting receive address register
  - Removed all NAHUM6LP_HW tags
- **Added e1000 RX interrupt support.**
- **Added igb TSO support for both PF and VF.**
- **Added RSS enhancements to Intel x550 NIC.**
  - Added support for 512 entry RSS redirection table.
  - Added support for per VF RSS redirection table.
- **Added Flow director enhancements on Intel x550 NIC.**
  - Added 2 new flow director modes on x550. One is MAC VLAN mode, the other is tunnel mode.
- **Updated the i40e base driver.**

  The i40e base driver was updated with several changes including the following:
  - Added promiscuous on VLAN support
  - Added a workaround to drop all flow control frames
  - Added VF capabilities to virtual channel interface
  - Added TX Scheduling related AQ commands
  - Added additional PCTYPES supported for FortPark RSS
  - Added parsing for CEE DCBX TLVs

  - Added FortPark specific registers
  - Added AQ functions to handle RSS Key and LUT programming
  - Increased PF reset max loop limit
- **Added i40e vector RX/TX.**
- **Added i40e RX interrupt support.**
- **Added i40e flow control support.**
- **Added DCB support to i40e PF driver.**
- **Added RSS/FD input set granularity on Intel X710/XL710.**
- **Added different GRE key length for input set on Intel X710/XL710.**
- **Added flow director support in i40e VF.**
- **Added i40e support of early X722 series.**

  Added early X722 support, for evaluation only, as the hardware is alpha.
- **Added fm10k vector RX/TX.**
- **Added fm10k TSO support for both PF and VF.**
- **Added fm10k VMDQ support.**
- **New NIC Boulder Rapid support.**

  Added support for the Boulder Rapid variant of Intel's fm10k NIC family.
- **Enhanced support for the Chelsio CXGBE driver.**
  - Added support for Jumbo Frames.
  - Optimized forwarding performance for Chelsio T5 40GbE cards.
- **Improved enic TX packet rate.**

  Reduced frequency of TX tail pointer updates to the NIC.
- **Added support for link status interrupts in mlx4.**
- **Added partial support (TX only) for secondary processes in mlx4.**
- **Added support for Mellanox ConnectX-4 adapters (mlx5).**

  The mlx5 poll-mode driver implements support for Mellanox ConnectX-4 EN and Mellanox ConnectX-4 Lx EN families of 10/25/40/50/100 Gb/s adapters.

  Like mlx4, this PMD is only available for Linux and is disabled by default due to external dependencies (libibverbs and libmlx5).
- **Added driver for Netronome nfp-6xxx card.**

  Support for using Netronome nfp-6xxx with PCI VFs.
- **Added virtual szedata2 driver for COMBO cards.**

  Added virtual PMD for COMBO-100G and COMBO-80G cards. PMD is disabled in default configuration.
- **Enhanced support for virtio driver.**

  - Virtio ring layout optimization (fixed avail ring)
  - Vector RX
  - Simple TX

- **Added vhost-user multiple queue support.**
- **Added port hotplug support to vmxnet3.**
- **Added port hotplug support to xenvirt.**
- **Added ethtool shim and sample application.**
- **Added experimental performance thread example application.**

  The new sample application demonstrates L3 forwarding with different threading models: pthreads, cgroups, or lightweight threads. The example includes a simple cooperative scheduler.

  Due to its experimental state this application may change without notice. The application is supported only for Linux x86_64.

- **Enhancements to the IP pipeline application.**

  The following features have been added to the `ip_pipeline` application;

  - Added Multiple Producers/Multiple Consumers (MPSC) and fragmentation/reassembly support to software rings.
  - Added a dynamic pipeline reconfiguration feature that allows binding a pipeline to other threads at runtime using CLI commands.
  - Added enable/disable of `promisc` mode from `ip_pipeline` configuration file.
  - Added check on RX queues and TX queues of each link whether they are used correctly in the `ip_pipeline` configuration file.
  - Added flow id parameters to the flow-classification table entries.
  - Added more functions to the routing pipeline: ARP table enable/disable, Q-in-Q and MPLS encapsulation, add color (traffic-class for QoS) to the MPLS tag.
  - Added flow-actions pipeline for traffic metering/marking (for e.g. Two Rate Three Color Marker (trTCM)), policer etc.
  - Modified the pass-through pipeline's actions-handler to implement a generic approach to extract fields from the packet's header and copy them to packet metadata.

5.2 Resolved Issues

5.2.1 EAL

- **eal/linux: Fixed epoll timeout.**

  Fixed issue where the `rte_epoll_wait()` function didn't return when the underlying call to `epoll_wait()` timed out.

5.2.2 Drivers

- **e1000/base: Synchronize PHY interface on non-ME systems.**

  On power up, the MAC - PHY interface needs to be set to PCIe, even if the cable is disconnected. In ME systems, the ME handles this on exit from the Sx (Sticky mode) state. In non-ME, the driver handles it. Added a check for non-ME system to the driver code that handles it.

- **e1000/base: Increased timeout of reset check.**

  Previously, in `check_reset_block` RSPCIPHY was polled for 100 ms before determining that the ME veto was set. This was not enough and it was increased to 300 ms.

- **e1000/base: Disabled IPv6 extension header parsing on 82575.**

  Disabled IPv6 options as per hardware limitation.

- **e1000/base: Prevent ULP flow if cable connected.**

  Enabling ULP on link down when the cable is connected caused an infinite loop of link up/down indications in the NDIS driver. The driver now enables ULP only when the cable is disconnected.

- **e1000/base: Support different EEARBC for i210.**

  EEARBC has changed on i210. It means EEARBC has a different address on i210 than on other NICs. So, add a new entity named EEARBC_I210 to the register list and make sure the right one is being used on i210.

- **e1000/base: Fix K1 configuration.**

  Added fix for the following updates to the K1 configurations: TX idle period for entering K1 should be 128 ns. Minimum TX idle period in K1 should be 256 ns.

- **e1000/base: Fix link detect flow.**

  Fix link detect flow in case where auto-negotiate is not enabled, by calling `e1000_setup_copper_link_generic` instead of `e1000_phy_setup_autoneg`.

- **e1000/base: Fix link check for i354 M88E1112 PHY.**

  The `e1000_check_for_link_media_swap()` function is supposed to check PHY page 0 for copper and PHY page 1 for "other" (fiber) links. The driver switched back from page 1 to page 0 too soon, before `e1000_check_for_link_82575()` is executed and was never finding the link on the fiber (other).

  If the link is copper, as the M88E1112 page address is set to 1, it should be set back to 0 before checking this link.

- **e1000/base: Fix beacon duration for i217.**

  Fix for I217 Packet Loss issue - The Management Engine sets the FEXTNVM4 Beacon Duration incorrectly. This fix ensures that the correct value will always be set. Correct value for this field is 8 usec.

- **e1000/base: Fix TIPG for non 10 half duplex mode.**

  TIPG value is increased when setting speed to 10 half duplex to prevent packet loss. However, it was never decreased again when speed changed. This caused performance issues in the NDIS driver. Fix this to restore TIPG to default value on non 10 half duplex.

- **e1000/base: Fix reset of DH89XXCC SGMII.**

  For DH89XXCC_SGMII, a write flush leaves registers of this device trashed (0xFFFFFFFF). Add check for this device.

  Also, after both Port SW Reset and Device Reset case, the platform should wait at least 3ms before reading any registers. Remove this condition since waiting is conditionally executed only for Device Reset.

- **e1000/base: Fix redundant PHY power down for i210.**

  Bit 11 of PHYREG 0 is used to power down PHY. The use of PHYREG 16 is no longer necessary.

- **e1000/base: fix jumbo frame CRC failures.**

  Change the value of register 776.20[11:2] for jumbo mode from 0x1A to 0x1F. This is to enlarge the gap between read and write pointers in the TX FIFO.

- **e1000/base: Fix link flap on 82579.**

  Several customers have reported a link flap issue on 82579. The symptoms are random and intermittent link losses when 82579 is connected to specific switches. the Issue was root caused as an inter-operability problem between the NIC and at least some Broadcom PHYs in the Energy Efficient Ethernet wake mechanism.

  To fix the issue, we are disabling the Phase Locked Loop shutdown in 100M Low Power Idle. This solution will cause an increase of power in 100M EEE link. It may cost an additional 28mW in this specific mode.

- **igb: Fixed IEEE1588 frame identification in I210.**

  Fixed issue where the flag `PKT_RX_IEEE1588_PTP` was not being set in the Intel I210 NIC, as the EtherType in RX descriptor is in bits 8:10 of Packet Type and not in the default bits 0:2.

- **igb: Fixed VF start with PF stopped.**

  VF needs the PF interrupt support initialized even if not started.

- **igb: Fixed VF MAC address when using with DPDK PF.**

  Assign a random MAC address in VF when not assigned by PF.

- **igb: Removed CRC bytes from byte counter statistics.**

- **ixgbe: Fixed issue with X550 DCB.**

  Fixed a DCB issue with x550 where for 8 TCs (Traffic Classes), if a packet with user priority 6 or 7 was injected to the NIC, then the NIC would only put 3 packets into the queue. There was also a similar issue for 4 TCs.

- **ixgbe: Removed burst size restriction of vector RX.**

  Fixed issue where a burst size less than 32 didn't receive anything.

- **ixgbe: Fixed VF start with PF stopped.**

  VF needs the PF interrupt support initialized even if not started.

- **ixgbe: Fixed TX hang when RS distance exceeds HW limit.**

Fixed an issue where the TX queue can hang when a lot of highly fragmented packets have to be sent. As part of that fix, `tx_rs_thresh` for ixgbe PMD is not allowed to be greater then to 32 to comply with HW restrictions.

- **ixgbe: Fixed rx error statistic counter.**

  Fixed an issue that the rx error counter of ixgbe was not accurate. The mac short packet discard count (mspdc) was added to the counter. Mac local faults and mac remote faults are removed as they do not count packets but errors, and jabber errors were removed as they are already accounted for by the CRC error counter. Finally the XEC (l3 / l4 checksum error) counter was removed due to errata, see commit 256ff05a9cae for details.

- **ixgbe: Removed CRC bytes from byte counter statistics.**
- **i40e: Fixed base driver allocation when not using first numa node.**

  Fixed i40e issue that occurred when a DPDK application didn't initialize ports if memory wasn't available on socket 0.

- **i40e: Fixed maximum of 64 queues per port.**

  Fixed an issue in i40e where it would not support more than 64 queues per port, even though the hardware actually supports it. The real number of queues may vary, as long as the total number of queues used in PF, VFs, VMDq and FD does not exceeds the hardware maximum.

- **i40e: Fixed statistics of packets.**

  Added discarding packets on VSI to the stats and rectify the old statistics.

- **i40e: Fixed issue of not freeing memzone.**

  Fixed an issue of not freeing a memzone in the call to free the memory for adminq DMA.

- **i40e: Removed CRC bytes from byte counter statistics.**
- **mlx: Fixed driver loading.**

  The mlx drivers were unable to load when built as a shared library, due to a missing symbol in the mempool library.

- **mlx4: Performance improvements.**

  Fixed bugs in TX and RX flows that improves mlx4 performance.

- **mlx4: Fixed TX loss after initialization.**
- **mlx4: Fixed scattered TX with too many segments.**
- **mlx4: Fixed memory registration for indirect mbuf data.**
- **vhost: Fixed Qemu shutdown.**

  Fixed issue with libvirt `virsh destroy` not killing the VM.

- **virtio: Fixed crash after changing link state.**

  Fixed IO permission in the interrupt handler.

- **virtio: Fixed crash when releasing queue.**

  Fixed issue when releasing null control queue.

5.2.3 Libraries

- **hash: Fixed memory allocation of Cuckoo Hash key table.**

  Fixed issue where an incorrect Cuckoo Hash key table size could be calculated limiting the size to 4GB.

- **hash: Fixed incorrect lookup if key is all zero.**

  Fixed issue in hash library that occurred if an all zero key was not added to the table and the key was looked up, resulting in an incorrect hit.

- **hash: Fixed thread scaling by reducing contention.**

  Fixed issue in the hash library where, using multiple cores with hardware transactional memory support, thread scaling did not work, due to the global ring that is shared by all cores.

5.2.4 Examples

- **l3fwd: Fixed crash with IPv6.**
- **vhost_xen: Fixed compile error.**

5.2.5 Other

- This release drops compatibility with Linux kernel 2.6.33. The minimum kernel requirement is now 2.6.34.

5.3 Known Issues

- Some drivers do not fill in the packet type when receiving. As the l3fwd example application requires this info, the i40e vector driver must be disabled to benefit of the packet type with i40e.
- Some (possibly all) VF drivers (e.g. i40evf) do not handle any PF reset events/requests in the VF driver. This means that the VF driver may not work after a PF reset in the host side. The workaround is to avoid triggering any PF reset events/requests on the host side.
- 100G link report support is missing.
- **Mellanox PMDs (mlx4 & mlx5):**
  - PMDs do not support CONFIG_RTE_BUILD_COMBINE_LIBS and CONFIG_RTE_BUILD_SHARED_LIB simultaneously.
  - There is performance degradation for small packets when the PMD is compiled with `SGE_WR_N = 4` compared to the performance when `SGE_WR_N = 1`. If scattered packets are not used it is recommended to compile the PMD with `SGE_WR_N = 1`.
  - When a Multicast or Broadcast packet is sent to the SR-IOV mlx4 VF, it is returned back to the port.
  - PMDs report “bad” L4 checksum when IP packet is received.

- mlx5 PMD reports "bad" checksum although the packet has "good" checksum. Will be fixed in upcoming MLNX_OFED release.

5.4 API Changes

- The deprecated flow director API is removed. It was replaced by rte_eth_dev_filter_ctrl().
- The dcb_queue is renamed to dcb_tc in following dcb configuration structures: rte_eth_dcb_rx_conf, rte_eth_dcb_tx_conf, rte_eth_vmdq_dcb_conf, rte_eth_vmdq_dcb_tx_conf.
- The rte_eth_rx_queue_count() function now returns "int" instead of "uint32_t" to allow the use of negative values as error codes on return.
- The function rte_eal_pci_close_one() is removed. It was replaced by rte_eal_pci_detach().
- The deprecated ACL API ipv4vlan is removed.
- The deprecated hash function rte_jhash2() is removed. It was replaced by rte_jhash_32b().
- The deprecated KNI functions are removed: rte_kni_create(), rte_kni_get_port_id() and rte_kni_info_get().
- The deprecated ring PMD functions are removed: rte_eth_ring_pair_create() and rte_eth_ring_pair_attach().
- The devargs union field virtual is renamed to virt for C++ compatibility.

5.5 ABI Changes

- The EAL and ethdev structures rte_intr_handle and rte_eth_conf were changed to support RX interrupt. This was already included in 2.1 under the CONFIG_RTE_NEXT_ABI #define.
- The ethdev flow director entries for SCTP were changed. This was already included in 2.1 under the CONFIG_RTE_NEXT_ABI #define.
- The ethdev flow director structure rte_eth_fdir_flow_ext structure was changed. New fields were added to support flow director filtering in VF.
- The size of the ethdev structure rte_eth_hash_filter_info is changed by adding a new element rte_eth_input_set_conf in a union.
- New fields rx_desc_lim and tx_desc_lim are added into rte_eth_dev_info structure.
- For debug builds, the functions rte_eth_rx_burst(), rte_eth_tx_burst() rte_eth_rx_descriptor_done() and rte_eth_rx_queue_count() will no longer be separate functions in the DPDK libraries. Instead, they will only be present in the rte_ethdev.h header file.
- The maximum number of queues per port CONFIG_RTE_MAX_QUEUES_PER_PORT is increased to 1024.

- The mbuf structure was changed to support the unified packet type. This was already included in 2.1 under the `CONFIG_RTE_NEXT_ABI` #define.
- The dummy malloc library is removed. The content was moved into EAL in 2.1.
- The LPM structure is changed. The deprecated field `mem_location` is removed.
- librte_table LPM: A new parameter to hold the table name will be added to the LPM table parameter structure.
- librte_table hash: The key mask parameter is added to the hash table parameter structure for 8-byte key and 16-byte key extendable bucket and LRU tables.
- librte_port: Macros to access the packet meta-data stored within the packet buffer has been adjusted to cover the packet mbuf structure.
- librte_cfgfile: Allow longer names and values by increasing the constants `CFG_NAME_LEN` and `CFG_VALUE_LEN` to 64 and 256 respectively.
- vhost: a new field enabled is added to the `vhost_virtqueue` structure.
- vhost: a new field `virt_qp_nb` is added to `virtio_net` structure, and the `virtqueue` field is moved to the end of virtio_net structure.
- vhost: a new operation `vring_state_changed` is added to `virtio_net_device_ops` structure.
- vhost: a few spaces are reserved both at `vhost_virtqueue` and `virtio_net` structure for future extension.

5.6 Shared Library Versions

The libraries prepended with a plus sign were incremented in this version.

```
+ libethdev.so.2
+ librte_acl.so.2
+ librte_cfgfile.so.2
  librte_cmdline.so.1
  librte_distributor.so.1
+ librte_eal.so.2
+ librte_hash.so.2
  librte_ip_frag.so.1
  librte_ivshmem.so.1
  librte_jobstats.so.1
+ librte_kni.so.2
  librte_kvargs.so.1
+ librte_lpm.so.2
+ librte_mbuf.so.2
  librte_mempool.so.1
  librte_meter.so.1
+ librte_pipeline.so.2
  librte_pmd_bond.so.1
+ librte_pmd_ring.so.2
+ librte_port.so.2
  librte_power.so.1
  librte_reorder.so.1
  librte_ring.so.1
  librte_sched.so.1
+ librte_table.so.2
  librte_timer.so.1
+ librte_vhost.so.2
```

CHAPTER SIX

DPDK RELEASE 2.1

6.1 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.

6.2 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.**

6.3 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.

6.4 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.

6.5 ABI Changes

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

CHAPTER

SEVEN

DPDK RELEASE 2.0

7.1 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

EIGHT

DPDK RELEASE 1.8

8.1 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 Intel 40GbE Controllers
- Packet Distributor Sample Application

CHAPTER

NINE

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

TEN

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.

10.1 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.

10.2 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).

10.3 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).

10.4 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`.

10.5 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).

10.6 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`

10.7 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 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

10.8 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 DPDK Programmer’s Guide for more information.

Affected Environment/Platform: All.

Driver/Module: Multi-process example application

10.9 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).

10.10 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).

10.11 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).

10.12 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).

10.13 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).

10.14 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).

10.15 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).

10.16 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).

10.17 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.

10.18 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).

10.19 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

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

Description: When compiling 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='-O3'` prior to compilation.

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

Driver/Module: Environment Abstraction Layer (EAL).

10.21 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).

10.22 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).

10.23 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.

10.24 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.

10.25 DPDK may not build on some Intel CPUs using clang < 3.7.0

Description: When compiling DPDK with an earlier version than 3.7.0 of clang, CPU flags are not detected on some Intel platforms such as Intel Broadwell/Skylake (and possibly future CPUs), and therefore compilation fails due to missing intrinsics.

Implication: DPDK will not build when using a clang version < 3.7.0.

Resolution/Workaround: Use clang 3.7.0 or higher, or gcc.

Affected Environment/Platform: Platforms with Intel Broadwell/Skylake using an old clang version.

Driver/Module: Environment Abstraction Layer (EAL).

10.26 The last EAL argument is replaced by the program name in argv[]

Description: The last EAL argument is replaced by program name in `argv[]` after `eal_parse_args` is called. This is the intended behavior but it causes the pointer to the last EAL argument to be lost.

Implication: If the last EAL argument in `argv[]` is generated by a malloc function, changing it will cause memory issues when freeing the argument.

Resolution/Workaround: An application should not consider the value in `argv[]` as unchanged.

Affected Environment/Platform: ALL.

Driver/Module: Environment Abstraction Layer (EAL).

10.27 I40e VF may not receive packets in the promiscuous mode

Description: Promiscuous mode is not supported by the DPDK i40e VF driver when using the i40e Linux kernel driver as host driver.

Implication: The i40e VF does not receive packets when the destination MAC address is unknown.

Resolution/Workaround: Use a explicit destination MAC address that matches the VF.

Affected Environment/Platform: All.

Driver/Module: Poll Mode Driver (PMD).

CHAPTER

ELEVEN

ABI AND API DEPRECATION

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

11.1 Deprecation Notices

- igb_uio: iomem mapping and sysfs files created for iomem and ioport in igb_uio will be removed, because we are able to detect these from what Linux has exposed, like the way we have done with uio-pci-generic. This change targets release 17.02.
- ABI/API changes are planned for 17.02: `rte_device`, `rte_driver` will be impacted because of introduction of a new `rte_bus` hierarchy. This would also impact the way devices are identified by EAL. A bus-device-driver model will be introduced providing a hierarchical view of devices.
- `eth_driver` is planned to be removed in 17.02. This currently serves as a placeholder for PMDs to register themselves. Changes for `rte_bus` will provide a way to handle device initialization currently being done in `eth_driver`.
- In 17.02 ABI changes are planned: the `rte_eth_dev` structure will be extended with new function pointer `tx_pkt_prepare` allowing verification and processing of packet burst to meet HW specific requirements before transmit. Also new fields will be added to the `rte_eth_desc_lim` structure: `nb_seg_max` and `nb_mtu_seg_max` providing information about number of segments limit to be transmitted by device for TSO/non-TSO packets.
- In 17.02 ABI change is planned: the `rte_eth_dev_info` structure will be extended with a new member `fw_version` in order to store the NIC firmware version.
- ethdev: an API change is planned for 17.02 for the function `_rte_eth_dev_callback_process`. In 17.02 the function will return an `int` instead of `void` and a fourth parameter `void *ret_param` will be added.
- ethdev: for 17.02 it is planned to deprecate the following five functions and move them in ixgbe:

  `rte_eth_dev_set_vf_rxmode`

  `rte_eth_dev_set_vf_rx`

  `rte_eth_dev_set_vf_tx`

  `rte_eth_dev_set_vf_vlan_filter`

rte_eth_set_vf_rate_limit

- ABI changes are planned for 17.02 in the rte_mbuf structure: some fields may be reordered to facilitate the writing of data_off, refcnt, and nb_segs in one operation, because some platforms have an overhead if the store address is not naturally aligned. Other mbuf fields, such as the port field, may be moved or removed as part of this mbuf work. A timestamp will also be added.
- The mbuf flags PKT_RX_VLAN_PKT and PKT_RX_QINQ_PKT are deprecated and are respectively replaced by PKT_RX_VLAN_STRIPPED and PKT_RX_QINQ_STRIPPED, that are better described. The old flags and their behavior will be kept until 16.11 and will be removed in 17.02.
- mempool: The functions rte_mempool_count and rte_mempool_free_count will be removed in 17.02. They are replaced by rte_mempool_avail_count and rte_mempool_in_use_count respectively.
- mempool: The functions for single/multi producer/consumer are deprecated and will be removed in 17.02. It is replaced by rte_mempool_generic_get/put functions.