Release Notes

Release 2.0.0

April 24, 2015

CONTENTS

Package Version: 2.0

April 24, 2015

Contents

CHAPTER

ONE

DESCRIPTION OF RELEASE

These release notes cover the new features, fixed bugs and known issues for Data Plane Development Kit (DPDK) release version 2.0.0.

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

1.1 Using DPDK Upgrade Patches

For minor updates to the main DPDK releases, the software may be made available both as a new full package and as a patch file to be applied to the previously released package. In the latter case, the following commands should be used to apply the patch on top of the already-installed package for the previous release:

```
# cd $RTE_SDK
# patch -p1 < /path/to/patch/file
```

Once the patch has been applied cleanly, the DPDK can be recompiled and used as before (described in the *DPDK Getting Started Guide*).

Note: If the patch does not apply cleanly, perhaps because of modifications made locally to the software, it is recommended to use the full release package for the minor update, instead of using the patch.

1.2 Documentation Roadmap

The following is a list of DPDK documents in the suggested reading order:

- **Release Notes** (this document): Provides release-specific information, including supported features, limitations, fixed issues, known issues and so on. Also, provides the answers to frequently asked questions in FAQ format.
- **Getting Started Guide** : Describes how to install and configure the DPDK software; designed to get users up and running quickly with the software.
- **FreeBSD* Getting Started Guide** : A document describing the use of the DPDK with FreeBSD* has been added in DPDK Release 1.6.0. Refer to this guide for installation and configuration instructions to get started using the DPDK with FreeBSD*.
- **Programmer's Guide** : Describes:

  - The software architecture and how to use it (through examples), specifically in a Linux* application (linuxapp) environment
  - The content of the DPDK, the build system (including the commands that can be used in the root DPDK Makefile to build the development kit and an application) and guidelines for porting an application
  - Optimizations used in the software and those that should be considered for new development

  A glossary of terms is also provided.

- **API Reference** : Provides detailed information about DPDK functions, data structures and other programming constructs.
- **Sample Applications User Guide** : Describes a set of sample applications. Each chapter describes a sample application that showcases specific functionality and provides instructions on how to compile, run and use the sample application.

  The following sample applications are included:

  - Command Line
  - Exception Path (into Linux* for packets using the Linux TUN/TAP driver)
  - Hello World
  - Integration with Intel® QuickAssist Technology
  - Link Status Interrupt (Ethernet* Link Status Detection)
  - IP Reassembly
  - IP Pipeline
  - IP Fragmentation
  - IPv4 Multicast
  - L2 Forwarding (supports virtualized and non-virtualized environments)
  - L2 Forwarding IVSHMEM
  - L2 Forwarding Jobstats
  - L3 Forwarding
  - L3 Forwarding with Access Control
  - L3 Forwarding with Power Management
  - L3 Forwarding in a Virtualized Environment
  - Link Bonding
  - Link Status Interrupt
  - Load Balancing
  - Multi-process
  - QoS Scheduler + Dropper
  - QoS Metering
  - Quota & Watermarks

- Timer
- VMDQ and DCB L2 Forwarding
- VMDQ L2 Forwarding
- Userspace vhost
- Userspace vhost switch
- Netmap
- Kernel NIC Interface (KNI)
- VM Power Management
- Distributor
- RX-TX Callbacks
- Skeleton

In addition, there are some other applications that are built when the libraries are created. The source for these applications is in the DPDK/app directory and are called:

- test
- testpmd

Once the libraries are created, they can be found in the build/app directory.

- The test application provides a variety of specific tests for the various functions in the DPDK.
- The testpmd application provides a number of different packet throughput tests and examples of features such as how to use the Flow Director found in the Intel® 82599 10 Gigabit Ethernet Controller.

The testpmd application is documented in the *DPDK Testpmd Application Note*. The test application is not currently documented. However, you should be able to run and use test application with the command line help that is provided in the application.

CHAPTER

TWO

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
- Packet Distributor Sample Application
- Job Stats library and Sample Application.

For further features supported in this release, see Chapter 3 Supported Features.

CHAPTER

THREE

SUPPORTED FEATURES

- Packet Distributor library for dynamic, single-packet at a time, load balancing
- IP fragmentation and reassembly library
- Support for IPv6 in IP fragmentation and reassembly sample applications
- Support for VFIO for mapping BARs and setting up interrupts
- Link Bonding PMD Library supporting round-robin, active backup, balance(layer 2, layer 2+3, and layer 3+4), broadcast bonding modes 802.3ad link aggregation (mode 4), transmit load balancing (mode 5) and adaptive load balancing (mode 6)
- Support zero copy mode RX/TX in user space vhost sample
- Support multiple queues in virtio-net PMD
- Support for Intel 40GbE Controllers:
  - Intel® XL710 40 Gigabit Ethernet Controller
  - Intel® X710 40 Gigabit Ethernet Controller
- Support NIC filters in addition to flow director for Intel® 1GbE and 10GbE Controllers
- Virtualization (KVM)
  - Userspace vhost switch:

    New sample application to support userspace virtio back-end in host and packet switching between guests.
- Virtualization (Xen)
  - Support for DPDK application running on Xen Domain0 without hugepages.
  - Para-virtualization

    Support front-end Poll Mode Driver in guest domain

    Support userspace packet switching back-end example in host domain
- FreeBSD* 9.2 support for librte_pmd_e1000, librte_pmd_ixgbe and Virtual Function variants. Please refer to the *DPDK for FreeBSD* Getting Started Guide*. Application support has been added for the following:
  - multiprocess/symmetric_mp
  - multiprocess/simple_mp
  - l2fwd

  - l3fwd
- Support for sharing data over QEMU IVSHMEM
- Support for Intel® Communications Chipset 8925 to 8955 Series in the DPDK-QAT Sample Application
- New VMXNET3 driver for the paravirtual device presented to a VM by the VMware* ESXi Hypervisor.
- BETA: example support for basic Netmap applications on DPDK
- Support for the wireless KASUMI algorithm in the dpdk_qat sample application
- Hierarchical scheduler implementing 5-level scheduling hierarchy (port, sub-port, pipe, traffic class, queue) with 64K leaf nodes (packet queues).
- Packet dropper based on Random Early Detection (RED) congestion control mechanism.
- Traffic Metering based on Single Rate Three Color Marker (srTCM) and Two Rate Three Color Marker (trTCM).
- An API for configuring RSS redirection table on the fly
- An API to support KNI in a multi-process environment
- IPv6 LPM forwarding
- Power management library and sample application using CPU frequency scaling
- IPv4 reassembly sample application
- Quota & Watermarks sample application
- PCIe Multi-BAR Mapping Support
- Support for Physical Functions in Poll Mode Driver for the following devices:
  - Intel® 82576 Gigabit Ethernet Controller
  - Intel® i350 Gigabit Ethernet Controller
  - Intel® 82599 10-Gigabit Ethernet Controller
  - Intel® XL710/X710 40-Gigabit Ethernet Controller
- Quality of Service (QoS) Hierarchical Scheduler: Sub-port Traffic Class Oversubscription
- Multi-thread Kernel NIC Interface (KNI) for performance improvement
- Virtualization (KVM)
  - Para-virtualization

    Support virtio front-end poll mode driver in guest virtual machine Support vHost raw socket interface as virtio back-end via KNI
  - SR-IOV Switching for the 10G Ethernet Controller

    Support Physical Function to start/stop Virtual Function Traffic

    Support Traffic Mirroring (Pool, VLAN, Uplink and Downlink)

    Support VF multiple MAC addresses (Exact/Hash match), VLAN filtering

    Support VF receive mode configuration

- Support VMDq for 1 GbE and 10 GbE NICs
- Extension for the Quality of Service (QoS) sample application to allow statistics polling
- New libpcap -based poll-mode driver, including support for reading from 3rd Party NICs using Linux kernel drivers
- New multi-process example using fork() to demonstrate application resiliency and recovery, including reattachment to and re-initialization of shared data structures where necessary
- New example (vmdq) to demonstrate VLAN-based packet filtering
- Improved scalability for scheduling large numbers of timers using the rte_timer library
- Support for building the DPDK as a shared library
- Support for Intel® Ethernet Server Bypass Adapter X520-SR2
- Poll Mode Driver support for the Intel® Ethernet Connection I354 on the Intel® Atom™ Processor C2000 Product Family SoCs
- IPv6 exact match flow classification in the l3fwd sample application
- Support for multiple instances of the Intel® DPDK
- Support for Intel® 82574L Gigabit Ethernet Controller - Intel® Gigabit CT Desktop Adapter (previously code named “Hartwell”)
- Support for Intel® Ethernet Controller I210 (previously code named “Springville”)
- Early access support for the Quad-port Intel® Ethernet Server Adapter X520-4 and X520-DA2 (code named “Spring Fountain”)
- Support for Intel® X710/XL710 40 Gigabit Ethernet Controller (code named “Fortville”)
- Core components:
  - rte_mempool: allocator for fixed-sized objects
  - rte_ring: single- or multi- consumer/producer queue implementation
  - rte_timer: implementation of timers
  - rte_malloc: malloc-like allocator
  - rte_mbuf: network packet buffers, including fragmented buffers
  - rte_hash: support for exact-match flow classification in software
  - rte_lpm: support for longest prefix match in software for IPv4 and IPv6
  - rte_sched: support for QoS scheduling
  - rte_meter: support for QoS traffic metering
  - rte_power: support for power management
  - rte_ip_frag: support for IP fragmentation and reassembly
- Poll Mode Driver - Common (rte_ether)
  - VLAN support
  - Support for Receive Side Scaling (RSS)

  - IEEE1588
  - Buffer chaining; Jumbo frames
  - TX checksum calculation
  - Configuration of promiscuous mode, and multicast packet receive filtering
  - L2 Mac address filtering
  - Statistics recording
- IGB Poll Mode Driver - 1 GbE Controllers (librte_pmd_e1000)
  - Support for Intel® 82576 Gigabit Ethernet Controller (previously code named "Kawela")
  - Support for Intel® 82580 Gigabit Ethernet Controller (previously code named "Barton Hills")
  - Support for Intel® I350 Gigabit Ethernet Controller (previously code named "Powerville")
  - Support for Intel® 82574L Gigabit Ethernet Controller - Intel® Gigabit CT Desktop Adapter (previously code named "Hartwell")
  - Support for Intel® Ethernet Controller I210 (previously code named "Springville")
  - Support for L2 Ethertype filters, SYN filters, 2-tuple filters and Flex filters for 82580 and i350
  - Support for L2 Ethertype filters, SYN filters and L3/L4 5-tuple filters for 82576
- Poll Mode Driver - 10 GbE Controllers (librte_pmd_ixgbe)
  - Support for Intel® 82599 10 Gigabit Ethernet Controller (previously code named "Niantic")
  - Support for Intel® Ethernet Server Adapter X520-T2 (previously code named "Iron Pond")
  - Support for Intel® Ethernet Controller X540-T2 (previously code named "Twin Pond")
  - Support for Virtual Machine Device Queues (VMDq) and Data Center Bridging (DCB) to divide incoming traffic into 128 RX queues. DCB is also supported for transmitting packets.
  - Support for auto negotiation down to 1 Gb
  - Support for Flow Director
  - Support for L2 Ethertype filters, SYN filters and L3/L4 5-tuple filters for 82599EB
- Poll Mode Driver - 40 GbE Controllers (librte_pmd_i40e)
  - Support for Intel® XL710 40 Gigabit Ethernet Controller
  - Support for Intel® X710 40 Gigabit Ethernet Controller
- Environment Abstraction Layer (librte_eal)
  - Multi-process support
  - Multi-thread support

  - 1 GB and 2 MB page support
  - Atomic integer operations
  - Querying CPU support of specific features
  - High Precision Event Timer support (HPET)
  - PCI device enumeration and blacklisting
  - Spin locks and R/W locks
- Test PMD application
  - Support for PMD driver testing
- Test application
  - Support for core component tests
- Sample applications
  - Command Line
  - Exception Path (into Linux* for packets using the Linux TUN/TAP driver)
  - Hello World
  - Integration with Intel® Quick Assist Technology drivers 1.0.0, 1.0.1 and 1.1.0 on Intel® Communications Chipset 89xx Series C0 and C1 silicon.
  - Link Status Interrupt (Ethernet* Link Status Detection
  - IPv4 Fragmentation
  - IPv4 Multicast
  - IPv4 Reassembly
  - L2 Forwarding (supports virtualized and non-virtualized environments)
  - L2 Forwarding Job Stats
  - L3 Forwarding (IPv4 and IPv6)
  - L3 Forwarding in a Virtualized Environment
  - L3 Forwarding with Power Management
  - Bonding mode 6
  - QoS Scheduling
  - QoS Metering + Dropper
  - Quota & Watermarks
  - Load Balancing
  - Multi-process
  - Timer
  - VMDQ and DCB L2 Forwarding
  - Kernel NIC Interface (with ethtool support)
  - Userspace vhost switch

- Interactive command line interface (rte_cmdline)
- Updated 10 GbE Poll Mode Driver (PMD) to the latest BSD code base providing support of newer ixgbe 10 GbE devices such as the Intel® X520-T2 server Ethernet adapter
- An API for configuring Ethernet flow control
- Support for interrupt-based Ethernet link status change detection
- Support for SR-IOV functions on the Intel® 82599, Intel® 82576 and Intel® i350 Ethernet Controllers in a virtualized environment
- Improvements to SR-IOV switch configurability on the Intel® 82599 Ethernet Controllers in a virtualized environment.
- An API for L2 Ethernet Address “whitelist” filtering
- An API for resetting statistics counters
- Support for RX L4 (UDP/TCP/SCTP) checksum validation by NIC
- Support for TX L3 (IPv4/IPv6) and L4 (UDP/TCP/SCTP) checksum calculation offloading
- Support for IPv4 packet fragmentation and reassembly
- Support for zero-copy Multicast
- New APIs to allow the “blacklisting” of specific NIC ports.
- Header files for common protocols (IP, SCTP, TCP, UDP)
- Improved multi-process application support, allowing multiple co-operating DPDK processes to access the NIC port queues directly.
- CPU-specific compiler optimization
- Job stats library for load/cpu utilization measurements
- Improvements to the Load Balancing sample application
- The addition of a PAUSE instruction to tight loops for energy-usage and performance improvements
- Updated 10 GbE Transmit architecture incorporating new upstream PCIe* optimizations.
- IPv6 support:
  - Support in Flow Director Signature Filters and masks
  - RSS support in sample application that use RSS
  - Exact match flow classification in the L3 Forwarding sample application
  - Support in LPM for IPv6 addresses
- Tunneling packet support:
  - Provide the APIs for VXLAN destination UDP port and VXLAN packet filter configuration and support VXLAN TX checksum offload on Intel® 40GbE Controllers.

CHAPTER

FOUR

SUPPORTED OPERATING SYSTEMS

The following Linux* distributions were successfully used to generate 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 *DPDK Getting Started Guide* for details.

CHAPTER

FIVE

UPDATING APPLICATIONS FROM PREVIOUS VERSIONS

Although backward compatibility is being maintained across DPDK releases, code written for previous versions of the DPDK may require some code updates to benefit from performance and user experience enhancements provided in later DPDK releases.

5.1 DPDK 1.7 to DPDK 1.8

Note that in DPDK 1.8, the structure of the rte_mbuf has changed considerably from all previous versions. It is recommended that users familiarize themselves with the new structure defined in the file rte_mbuf.h in the release package. The follow are some common changes that need to be made to code using mbufs, following an update to DPDK 1.8:

- Any references to fields in the pkt or ctrl sub-structures of the mbuf, need to be replaced with references to the field directly from the rte_mbuf, i.e. buf->pkt.data_len should be replace by buf->data_len.
- Any direct references to the data field of the mbuf (original buf->pkt.data) should now be replace by the macro rte_pktmbuf_mtod to get a computed data address inside the mbuf buffer area.
- Any references to the in_port mbuf field should be replace by references to the port field.

NOTE: The above list is not exhaustive, but only includes the most commonly required changes to code using mbufs.

5.2 Intel® DPDK 1.6 to DPDK 1.7

Note the following difference between 1.6 and 1.7:

- The “default” target has been renamed to “native”

5.3 Intel® DPDK 1.5 to Intel® DPDK 1.6

Note the following difference between 1.5 and 1.6:

- The CONFIG_RTE_EAL _UNBIND_PORTS configuration option, which was deprecated in Intel® DPDK 1.4.x, has been removed in Intel® DPDK 1.6.x. Applications using the Intel® DPDK must be explicitly unbound to the igb_uio driver using the dpdk_nic_bind.py

script included in the Intel® DPDK release and documented in the *Intel® DPDK Getting Started Guide*.

5.4 Intel® DPDK 1.4 to Intel® DPDK 1.5

Note the following difference between 1.4 and 1.5:

- Starting with version 1.5, the top-level directory created from unzipping the release package will now contain the release version number, that is, DPDK-1.5.2/ rather than just DPDK/ .

5.5 Intel® DPDK 1.3 to Intel® DPDK 1.4.x

Note the following difference between releases 1.3 and 1.4.x:

- In Release 1.4.x, Intel® DPDK applications will no longer unbind the network ports from the Linux* kernel driver when the application initializes. Instead, any ports to be used by Intel® DPDK must be unbound from the Linux driver and bound to the igb_uio driver before the application starts. This can be done using the pci_unbind.py script included with the Intel® DPDK release and documented in the *Intel® DPDK Getting Started Guide*.

  If the port unbinding behavior present in previous Intel® DPDK releases is required, this can be re-enabled using the CONFIG_RTE_EAL_UNBIND_PORTS setting in the appropriate Intel® DPDK compile-time configuration file.

- In Release 1.4.x, HPET support is disabled in the Intel® DPDK build configuration files, which means that the existing rte_eal_get_hpet_hz() and rte_eal_get_hpet_cycles() APIs are not available by default. For applications that require timing APIs, but not the HPET timer specifically, it is recommended that the API calls rte_get_timer_cycles() and rte_get_timer_hz() be used instead of the HPET-specific APIs. These generic APIs can work with either TSC or HPET time sources, depending on what is requested by an application, and on what is available on the system at runtime.

  For more details on this and how to re-enable the HPET if it is needed, please consult the *Intel® DPDK Getting Started Guide*.

5.6 Intel® DPDK 1.2 to Intel® DPDK 1.3

Note the following difference between releases 1.2 and 1.3:

- In release 1.3, the Intel® DPDK supports two different 1 GbE drivers: igb and em. Both of them are located in the same library: lib_pmd_e1000.a. Therefore, the name of the library to link with for the igb PMD has changed from librte_pmd_igb.a to librte_pmd_e1000.a.

- The rte_common.h macros, RTE_ALIGN, RTE_ALIGN_FLOOR and RTE_ALIGN_CEIL were renamed to, RTE_PTR_ALIGN, RTE_PTR_ALIGN_FLOOR and RTE_PTR_ALIGN_CEIL. The original macros are still available but they have different behavior. Not updating the macros results in strange compilation errors.

- The rte_tailq is now defined statically. The rte_tailq APIs have also been changed from being public to internal use only. The old public APIs are maintained for backward compatibility reasons. Details can be found in the *Intel® DPDK API Reference*.
- The method for managing mbufs on the NIC RX rings has been modified to improve performance. To allow applications to use the newer, more optimized, code path, it is recommended that the rx_free_thresh field in the rte_eth_conf structure, which is passed to the Poll Mode Driver when initializing a network port, be set to a value of 32.

5.7 Intel® DPDK 1.1 to Intel® DPDK 1.2

Note the following difference between release 1.1 and release 1.2:

- The names of the 1G and 10G Ethernet drivers have changed between releases 1.1 and 1.2. While the old driver names still work, it is recommended that code be updated to the new names, since the old names are deprecated and may be removed in a future release.

  The items affected are as follows:

  - Any macros referring to RTE_LIBRTE_82576_PMD should be updated to refer to RTE_LIBRTE_IGB_PMD.
  - Any macros referring to RTE_LIBRTE_82599_PMD should be updated to refer to RTE_LIBRTE_IXGBE_PMD.
  - Any calls to the rte_82576_pmd_init() function should be replaced by calls to rte_igb_pmd_init().
  - Any calls to the rte_82599_pmd_init() function should be replaced by calls to rte_ixgbe_pmd_init().
- The method used for managing mbufs on the NIC TX rings for the 10 GbE driver has been modified to improve performance. As a result, different parameter values should be passed to the rte_eth_tx_queue_setup() function. The recommended default values are to have tx_thresh.tx_wthresh, tx_free_thresh, as well as the new parameter tx_rs_thresh (all in the struct rte_eth_txconf datatype) set to zero. See the “Configuration of Transmit and Receive Queues” section in the *Intel® DPDK Programmer's Guide* for more details.

Note: If the tx_free_thresh field is set to TX_RING_SIZE+1 , as was previously used in some cases to disable free threshold check, then an error is generated at port initialization time. To avoid this error, configure the TX threshold values as suggested above.

CHAPTER

SIX

KNOWN ISSUES AND LIMITATIONS

This section describes known issues with the DPDK software.

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

Title	Unit Test for Link Bonding may fail at test_tlb_tx_burst()
Reference #	IXA00390304
Description	Unit tests will fail at test_tlb_tx_burst function with error for uneven distribution of packets.
Implication	Unit test link_bonding_autotest will fail
Resolution/ Workaround	There is no workaround available.
Affected Environment/ Platform	Fedora 20
Driver/Module	Link Bonding

6.2 Pause Frame Forwarding does not work properly on igb

Title	Pause Frame forwarding does not work properly on igb
Reference #	IXA00384637
Description	For igb devices rte_eth_flow_ctrl_set is not working as expected. Pause frames are always forwarded on igb, regardless of the RFCE, MPMCF and DPF registers.
Implication	Pause frames will never be rejected by the host on 1G NICs and they will always be forwarded.
Resolution/ Workaround	There is no workaround available.
Affected Environment/ Platform	All
Driver/Module	Poll Mode Driver (PMD)

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

Title	In packets provided by the PMD, some flags are missing
Reference #	3
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	The application has to parse the Ethernet header itself to get the information, which is slower.
Affected Environment/ Platform	All
Driver/Module	Poll Mode Driver (PMD)

6.4 The rte_malloc library is not fully implemented

Title	The rte_malloc library is not fully implemented
Reference #	6
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	No workaround available.
Affected Environment/ Platform	All
Driver/Module	rte_malloc

6.5 HPET reading is slow

Title	HPET reading is slow
Reference #	7
Description	Reading the HPET chip is slow.
Implication	An application that calls “rte_get_hpet_cycles()” or “rte_timer_manage()” runs slower.
Resolution	The application should not call these functions too often in the main loop. An alternative is to use the TSC register through “rte_rdtsc()” which is faster, but specific to an lcore and is a cycle reference, not a time reference.
Affected Environment/ Platform	All
Driver/Module	Environment Abstraction Layer (EAL)

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

Title	HPET timers do not work on the Osage customer reference platform
Reference #	17
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	This can be addressed by building the system with the “CONFIG_RTE_LIBEAL_USE_HPET=n” configuration option or by using the –no-hpet EAL option.
Affected Environment/ Platform	The Osage customer reference platform. Other vendor platforms with Intel® Xeon® processor 5500 series processors should work correctly, provided the BIOS supports HPET.
Driver/Module	lib/librte_eal/common/include/rte_cycles.h

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

Title	Not all variants of supported NIC types have been used in testing
Reference #	28
Description	The supported network interface cards can come in a number of variants with different device ID's. Not all of these variants have been tested with the Intel® DPDK. The NIC device identifiers used during testing: • Intel® Ethernet Controller XL710 for 40GbE QSFP+ [8086:1584] • Intel® Ethernet Controller XL710 for 40GbE QSFP+ [8086:1583] • Intel® Ethernet Controller X710 for 10GbE SFP+ [8086:1572] • Intel® 82576 Gigabit Ethernet Controller [8086:10c9] • Intel® 82576 Quad Copper Gigabit Ethernet Controller [8086:10e8] • Intel® 82580 Dual Copper Gigabit Ethernet Controller [8086:150e] • Intel® I350 Quad Copper Gigabit Ethernet Controller [8086:1521] • Intel® 82599 Dual Fibre 10 Gigabit Ethernet Controller [8086:10fb] • Intel® Ethernet Server Adapter X520-T2 [8086: 151c] • Intel® Ethernet Controller X540-T2 [8086:1528] • Intel® 82574L Gigabit Network Connection [8086:10d3] • Emulated Intel® 82540EM Gigabit Ethernet Controller [8086:100e] • Emulated Intel® 82545EM Gigabit Ethernet Controller [8086:100f] • Intel® Ethernet Server Adapter X520-4 [8086:154a] • Intel® Ethernet Controller I210 [8086:1533]
Implication	Risk of issues with untested variants.
Resolution	Use tested NIC variants. For those supported Ethernet controllers, additional device IDs may be added to the software if required.
Affected Environment/ Platform	All
Driver/Module	Poll-mode drivers

6.8 Multi-process sample app requires exact memory mapping

Title	Multi-process sample app requires exact memory mapping
Reference #	30
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	See the “Multi-process Limitations” section in the Intel® DPDK Programmer’s Guide for more information.
Affected Environment/ Platform	All
Driver/Module	Multi-process example application

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

Title	Packets are not sent by the 1 GbE/10 GbE SR-IOV driver when the source MAC address is not the MAC address assigned to the VF NIC
Reference #	IXA00168379
Description	The 1 GbE/10 GbE SR-IOV driver can only send packets when the Ethernet header’s source MAC address is the same as that of the VF NIC. The reason for this is that the Linux “ixgbe” driver module in the host OS has its anti-spoofing feature enabled.
Implication	Packets sent using the 1 GbE/10 GbE SR-IOV driver must have the source MAC address correctly set to that of the VF NIC. Packets with other source address values are dropped by the NIC if the application attempts to transmit them.
Resolution/ Workaround	Configure the Ethernet source address in each packet to match that of the VF NIC.
Affected Environment/ Platform	All
Driver/Module	1 GbE/10 GbE VF Poll Mode Driver (PMD)

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

Title	SR-IOV drivers do not fully implement the rte_ethdev API
Reference #	59
Description	The SR-IOV drivers only supports the following rte_ethdev API functions: • rte_eth_dev_configure() • rte_eth_tx_queue_setup() • rte_eth_rx_queue_setup() • rte_eth_dev_info_get() • rte_eth_dev_start() • rte_eth_tx_burst() • rte_eth_rx_burst() • rte_eth_dev_stop() • rte_eth_stats_get() • rte_eth_stats_reset() • rte_eth_link_get() • rte_eth_link_get_no_wait()
Implication	Calling an unsupported function will result in an application error.
Resolution/ Workaround	Do not use other rte_ethdev API functions in applications that use the SR-IOV drivers.
Affected Environment/ Platform	All
Driver/Module	VF Poll Mode Driver (PMD)

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

Title	PMD does not work with –no-huge EAL command line parameter
Reference #	IXA00373461
Description	Currently, the DPDK does not store any information about memory allocated by malloc() (for example, NUMA node, physical address), hence PMD drivers do not work when the –no-huge command line parameter is supplied to EAL.
Implication	Sending and receiving data with PMD will not work.
Resolution/ Workaround	Use huge page memory or use VFIO to map devices.
Affected Environment/ Platform	Systems running the DPDK on Linux
Driver/Module	Poll Mode Driver (PMD)

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

Title	Some hardware off-load functions are not supported by the VF Driver
Reference #	IXA00378813
Description	Currently, configuration of the following items is not supported by the VF driver: • IP/UDP/TCP checksum offload • Jumbo Frame Receipt • HW Strip CRC
Implication	Any configuration for these items in the VF register will be ignored. The behavior is dependent on the current PF setting.
Resolution/ Workaround	For the PF (Physical Function) status on which the VF driver depends, there is an option item under PMD in the config file. For others, the VF will keep the same behavior as PF setting.
Affected Environment/ Platform	All
Driver/Module	VF (SR-IOV) Poll Mode Driver (PMD)

6.13 Kernel crash on IGB port unbinding

Title	Kernel crash on IGB port unbinding
Reference #	74
Description	Kernel crash may occur when unbinding 1G ports from the igb_uio driver, on 2.6.3x kernels such as shipped with Fedora 14.
Implication	Kernel crash occurs.
Resolution/ Workaround	Use newer kernels or do not unbind ports.
Affected Environment/ Platform	2.6.3x kernels such as shipped with Fedora 14
Driver/Module	IGB Poll Mode Driver (PMD)

6.14 Twinpond and Ironpond NICs do not report link status correctly

Title	Twinpond and Ironpond NICs do not report link status correctly
Reference #	IXA00378800
Description	Twin Pond/Iron Pond NICs do not bring the physical link down when shutting down the port.
Implication	The link is reported as up even after issuing “shutdown” command unless the cable is physically disconnected.
Resolution/ Workaround	None.
Affected Environment/ Platform	Twin Pond and Iron Pond NICs
Driver/Module	Poll Mode Driver (PMD)

6.15 Discrepancies between statistics reported by different NICs

Title	Discrepancies between statistics reported by different NICs
Reference #	IXA00378113
Description	Gigabit Ethernet devices from Intel include CRC bytes when calculating packet reception statistics regardless of hardware CRC stripping state, while 10-Gigabit Ethernet devices from Intel do so only when hardware CRC stripping is disabled.
Implication	There may be a discrepancy in how different NICs display packet reception statistics.
Resolution/ Workaround	None
Affected Environment/ Platform	All
Driver/Module	Poll Mode Driver (PMD)

6.16 Error reported opening files on DPDK initialization

Title	Error reported opening files on DPDK initialization
Reference #	91
Description	On DPDK application startup, errors may be reported when opening files as part of the initialization process. This occurs if a large number, for example, 500 or more, or if hugepages are used, due to the per-process limit on the number of open files.
Implication	The DPDK application may fail to run.
Resolution/ Workaround	If using 2 MB hugepages, consider switching to a fewer number of 1 GB pages. Alternatively, use the “ulimit” command to increase the number of files which can be opened by a process.
Affected Environment/ Platform	All
Driver/Module	Environment Abstraction Layer (EAL)

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

Title	Intel® QuickAssist Technology sample applications does not work on a 32- bit OS on Shumway
Reference #	93
Description	The Intel® Communications Chipset 89xx Series device does not fully support NUMA on a 32-bit OS. Consequently, the sample application cannot work properly on Shumway, since it requires NUMA on both nodes.
Implication	The sample application cannot work in 32-bit mode with emulated NUMA, on multi-socket boards.
Resolution/ Workaround	There is no workaround available.
Affected Environment/ Platform	Shumway
Driver/Module	All

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

Title	IEEE1588 support may not work with an Intel® Ethernet Controller I210 NIC
Reference #	IXA00380285
Description	IEEE1588 support is not working with an Intel® Ethernet Controller I210 NIC.
Implication	IEEE1588 packets are not forwarded correctly by the Intel® Ethernet Controller I210 NIC.
Resolution/ Workaround	There is no workaround available.
Affected Environment/ Platform	All
Driver/Module	IGB Poll Mode Driver

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

Title	Differences in how different Intel NICs handle maximum packet length for jumbo frame
Reference #	96
Description	10 Gigabit Ethernet devices from Intel do not take VLAN tags into account when calculating packet size while Gigabit Ethernet devices do so for jumbo frames.
Implication	When receiving packets with VLAN tags, the actual maximum size of useful payload that Intel Gigabit Ethernet devices are able to receive is 4 bytes (or 8 bytes in the case of packets with extended VLAN tags) less than that of Intel 10 Gigabit Ethernet devices.
Resolution/ Workaround	Increase the configured maximum packet size when using Intel Gigabit Ethernet devices.
Affected Environment/ Platform	All
Driver/Module	Poll Mode Driver (PMD)

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

Title	Binding PCI devices to igb_uio fails on Linux* kernel 3.9 when more than one device is used
Reference #	97
Description	A known bug in the uio driver included in Linux* kernel version 3.9 prevents more than one PCI device to be bound to the igb_uio driver.
Implication	The Poll Mode Driver (PMD) will crash on initialization.
Resolution/ Workaround	Use earlier or later kernel versions, or apply the following patch .
Affected Environment/ Platform	Linux* systems with kernel version 3.9
Driver/Module	igb_uio module

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

Title	Gcc might generate Intel® AVX instructions for processors without Intel® AVX support
Reference #	IXA00382439
Description	When compiling Intel® DPDK (and any DPDK app), gcc may generate Intel® AVX instructions, even when the processor does not support Intel® AVX.
Implication	Any DPDK app might crash while starting up.
Resolution/ Workaround	Either compile using icc or set EXTRA_CFLAGS='-O3' prior to compilation.
Affected Environment/ Platform	Platforms which processor does not support Intel® AVX.
Driver/Module	Environment Abstraction Layer (EAL)

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

Title	Ethertype filter could receive other packets (non-assigned) in Niantic
Reference #	IXA00169017
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: “add_ethertype_filter 0 ethertype 0x0806 priority enable 3 queue 2 index 1”, and then start forwarding. 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 the NIC’s issue due to the response from PAE: “In fact, ETQF.UP is not functional, and the information will be added in errata of 82599 and X540.”
Resolution/ Workaround	None
Affected Environment/ Platform	All
Driver/Module	Poll Mode Driver (PMD)

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

Title	Cannot set link speed on Intel® 40G Ethernet controller
Reference #	IXA00386379
Description	On Intel® 40G Ethernet Controller: It cannot set the link to specific speed.
Implication	The link speed cannot be changed forcibly, though it can be configured by application.
Resolution/ Workaround	None
Affected Environment/ Platform	All
Driver/Module	Poll Mode Driver (PMD)

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

Title	Stopping the port does not down the link on Intel® 40G Ethernet controller
Reference #	IXA00386380
Description	On Intel® 40G Ethernet Controller: Stopping the port does not really down the port link.
Implication	The port link will be still up after stopping the port.
Resolution/ Workaround	None
Affected Environment/ Platform	All
Driver/Module	Poll Mode Driver (PMD)

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

Title	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. ie. 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

CHAPTER

SEVEN

RESOLVED ISSUES

This section describes previously known issues that have been resolved since release version 1.2.

7.1 Running TestPMD with SRIOV in Domain U may cause it to hang when XENVIRT switch is on

Title	Running TestPMD with SRIOV in Domain U may cause it to hang when XENVIRT switch is on
Reference #	IXA00168949
Description	When TestPMD is run with only SRIOV port /testpmd -c f -n 4 – -i, the following error occurs: PMD: gntalloc: ioctl error EAL: Error - exiting with code: 1 Cause: Creation of mbuf pool for socket 0 failed Then, alternately run SRIOV port and virtIO with testpmd: testpmd -c f -n 4 – -i testpmd -c f -n 4 –use-dev=”eth_xenvirt0” – -i
Implication	DomU will not be accessible after you repeat this action some times
Resolution/ Workaround	Run testpmd with a “–total-num-mbufs=N(N<=3500)”
Affected Environment/ Platform	Fedora 16, 64 bits + Xen hypervisor 4.2.3 + Domain 0 kernel 3.10.0 +Domain U kernel 3.6.11
Driver/Module	TestPMD Sample Application

7.2 Vhost-xen cannot detect Domain U application exit on Xen version 4.0.1

Title	Vhost-xen cannot detect Domain U application exit on Xen 4.0.1.
Reference #	IXA00168947
Description	When using DPDK applications on Xen 4.0.1, e.g. TestPMD Sample Application, on killing the application (e.g. killall testpmd) vhost-switch cannot detect the domain U exited and does not free the Virtio device.
Implication	**Virtio device not freed after application is killed when us** 4.0.1
Resolution	Resolved in DPDK 1.8
Affected Environment/ Platform	Xen 4.0.1
Driver/Module	Vhost-switch

7.3 Virtio incorrect header length used if MSI-X is disabled by kernel driver

Title	Virtio incorrect header length used if MSI-X is disabled by kernel driver or if VIRTIO_NET_F_MAC is not negotiated.
Reference #	IXA00384256
Description	The Virtio header for host-guest communication is of variable length and is dependent on whether MSI-X has been enabled by the kernel driver for the network device. The base header length of 20 bytes will be extended by 4 bytes to accommodate MSI-X vectors and the Virtio Network Device header will appear at byte offset 24. The Userspace Virtio Poll Mode Driver tests the guest feature bits for the presence of VIRTIO_PCI_FLAG_MISIX, however this bit field is not part of the Virtio specification and resolves to the VIRTIO_NET_F_MAC feature instead.
Implication	The DPDK kernel driver will enable MSI-X by default, however if loaded with “intr_mode=legacy” on a guest with a Virtio Network Device, a KVM-Qemu guest may crash with the following error: “virtio-net header not in first element”. If VIRTIO_NET_F_MAC feature has not been negotiated, then the Userspace Poll Mode Driver will assume that MSI-X has been disabled and will prevent the proper functioning of the driver.
Resolution	Ensure #define VIRTIO_PCI_CONFIG(hw) returns the correct offset (20 or 24 bytes) for the devices where in rare cases MSI-X is disabled or VIRTIO_NET_F_MAC has not been negotiated.
Affected Environment/ Platform	Virtio devices where MSI-X is disabled or VIRTIO_NET_F_MAC feature has not been negotiated.
Driver/Module	librte_pmd_virtio

7.4 Unstable system performance across application executions with 2MB pages

Title	Unstable system performance across application executions with 2MB pages
Reference #	IXA00372346
Description	The performance of an DPDK application may vary across executions of an application due to a varying number of TLB misses depending on the location of accessed structures in memory. This situation occurs on rare occasions.
Implication	Occasionally, relatively poor performance of DPDK applications is encountered.
Resolution/ Workaround	Using 1 GB pages results in lower usage of TLB entries, resolving this issue.
Affected Environment/ Platform	Systems using 2 MB pages
Driver/Module	All

7.5 Link status change not working with MSI interrupts

Title	Link status change not working with MSI interrupts
Reference #	IXA00378191
Description	MSI interrupts are not supported by the PMD.
Implication	Link status change will only work with legacy or MSI-X interrupts.
Resolution/ Workaround	The igb_uio driver can now be loaded with either legacy or MSI-X interrupt support. However, this configuration is not tested.
Affected Environment/ Platform	All
Driver/Module	Poll Mode Driver (PMD)

7.6 KNI does not provide Ethtool support for all NICs supported by the Poll-Mode Drivers

Title	KNI does not provide ethtool support for all NICs supported by the Poll Mode Drivers
Reference #	IXA00383835
Description	To support ethtool functionality using the KNI, the KNI library includes separate driver code based off the Linux kernel drivers, because this driver code is separate from the poll-mode drivers, the set of supported NICs for these two components may differ. Because of this, in this release, the KNI driver does not provide “ethtool” support for the Intel® Ethernet Connection I354 on the Intel Atom Processor C2000 product Family SoCs.
Implication	Ethtool support with KNI will not work for NICs such as the Intel® Ethernet Connection I354. Other KNI functionality, such as injecting packets into the Linux kernel is unaffected.
Resolution/Workaround	Updated for Intel® Ethernet Connection I354.
Affected Environment/Platform	Platforms using the Intel® Ethernet Connection I354 or other NICs unsupported by KNI ethtool
Driver/Module	KNI

7.7 Linux IPv4 forwarding is not stable with vhost-switch on high packet rate

Title	Linux IPv4 forwarding is not stable with vhost-switch on high packet rate.
Reference #	IXA00384430
Description	Linux IPv4 forwarding is not stable in Guest when Tx traffic is high from traffic generator using two virtio devices in VM with 10G in host.
Implication	Packets cannot be forwarded by user space vhost-switch and Linux IPv4 forwarding if the rate of incoming packets is greater than 1 Mpps.
Resolution/Workaround	N/A
Affected Environment/Platform	All
Driver/Module	Sample application

7.8 PCAP library overwrites mbuf data before data is used

Title	PCAP library overwrites mbuf data before data is used
Reference #	IXA00383976
Description	PCAP library allocates 64 mbufs for reading packets from PCAP file, but declares them as static and reuses the same mbufs repeatedly rather than handing off to the ring for allocation of new mbuf for each read from the PCAP file.
Implication	In multi-threaded applications ata in the mbuf is overwritten.
Resolution/Workaround	Fixed in eth_pcap_rx() in rte_eth_pcap.c
Affected Environment/Platform	All
Driver/Module	Multi-threaded applications using PCAP library

7.9 MP Client Example app - flushing part of TX is not working for some ports if set specific port mask with skipped ports

Title	MP Client Example app - flushing part of TX is not working for some ports if set specific port mask with skipped ports
Reference #	52
Description	When ports not in a consecutive set, for example, ports other than ports 0, 1 or 0,1,2,3 are used with the client-service sample app, when no further packets are received by a client, the application may not flush correctly any unsent packets already buffered inside it.
Implication	Not all buffered packets are transmitted if traffic to the clients application is stopped. While traffic is continually received for transmission on a port by a client, buffer flushing happens normally.
Resolution/Workaround	Changed line 284 of the client.c file: from “send_packets(ports);” to “send_packets(ports->id[port]);”
Affected Environment/Platform	All
Driver/Module	Client - Server Multi-process Sample application

7.10 Packet truncation with Intel® I350 Gigabit Ethernet Controller

Title	Packet truncation with Intel I350 Gigabit Ethernet Controller
Reference #	IXA00372461
Description	The setting of the hw_strip_crc field in the rte_eth_conf structure passed to the rte_eth_dev_configure() function is not respected and hardware CRC stripping is always enabled. If the field is set to 0, then the software also tries to strip the CRC, resulting in packet truncation.
Implication	The last 4 bytes of the packets received will be missing.
Resolution/Workaround	Fixed an omission in device initialization (setting the STRCRC bit in the DVMOLR register) to respect the CRC stripping selection correctly.
Affected Environment/Platform	Systems using the Intel® I350 Gigabit Ethernet Controller
Driver/Module	1 GbE Poll Mode Driver (PMD)

7.11 Device initialization failure with Intel® Ethernet Server Adapter X520-T2

Title	Device initialization failure with Intel® Ethernet Server Adapter X520-T2
Reference #	55
Description	If this device is bound to the Linux kernel IXGBE driver when the DPDK is initialized, DPDK is initialized, the device initialization fails with error code -17 “IXGBE_ERR_PHY_ADDR_INVALID”.
Implication	The device is not initialized and cannot be used by an application.
Resolution/Workaround	Introduced a small delay in device initialization to allow DPDK to always find the device.
Affected Environment/Platform	Systems using the Intel® Ethernet Server Adapter X520-T2
Driver/Module	10 GbE Poll Mode Driver (PMD)

7.12 DPDK kernel module is incompatible with Linux kernel version 3.3

Title	DPDK kernel module is incompatible with Linux kernel version 3.3
Reference #	IXA00373232
Description	The igb_uio kernel module fails to compile on systems with Linux kernel version 3.3 due to API changes in kernel headers
Implication	The compilation fails and Ethernet controllers fail to initialize without the igb_uio module.
Resolution/Workaround	Kernel functions pci_block_user_cfg_access() / pci_cfg_access_lock() and pci_unblock_user_cfg_access() / pci_cfg_access_unlock() are automatically selected at compile time as appropriate.
Affected Environment/Platform	Linux systems using kernel version 3.3 or later
Driver/Module	UIO module

7.13 Initialization failure with Intel® Ethernet Controller X540-T2

Title	Initialization failure with Intel® Ethernet Controller X540-T2
Reference #	57
Description	This device causes a failure during initialization when the software tries to read the part number from the device EPROM.
Implication	Device cannot be used.
Resolution/Workaround	Remove unnecessary check of the PBA number from the device.
Affected Environment/Platform	Systems using the Intel® Ethernet Controller X540-T2
Driver/Module	10 GbE Poll Mode Driver (PMD)

7.14 rte_eth_dev_stop() function does not bring down the link for 1 GB NIC ports

Title	rte_eth_dev_stop() function does not bring down the link for 1 GB NIC ports
Reference #	IXA00373183
Description	When the rte_eth_dev_stop() function is used to stop a NIC port, the link is not brought down for that port.
Implication	Links are still reported as up, even though the NIC device has been stopped and cannot perform TX or RX operations on that port.
Resolution	The rte_eth_dev_stop() function now brings down the link when called.
Affected Environment/Platform	All
Driver/Module	1 GbE Poll Mode Driver (PMD)

7.15 It is not possible to adjust the duplex setting for 1GB NIC ports

Title	It is not possible to adjust the duplex setting for 1 GB NIC ports
Reference #	66
Description	The rte_eth_conf structure does not have a parameter that allows a port to be set to half-duplex instead of full-duplex mode, therefore, 1 GB NICs cannot be configured explicitly to a full- or half-duplex value.
Implication	1 GB port duplex capability cannot be set manually.
Resolution	The PMD now uses a new field added to the rte_eth_conf structure to allow 1 GB ports to be configured explicitly as half- or full-duplex.
Affected Environment/Platform	All
Driver/Module	1 GbE Poll Mode Driver (PMD)

7.16 Calling rte_eth_dev_stop() on a port does not free all the mbufs in use by that port

Title	Calling rte_eth_dev_stop() on a port does not free all the mbufs in use by that port
Reference #	67
Description	The rte_eth_dev_stop() function initially frees all mbufs used by that port's RX and TX rings, but subsequently repopulates the RX ring again later in the function.
Implication	Not all mbufs used by a port are freed when the port is stopped.
Resolution	The driver no longer re-populates the RX ring in the rte_eth_dev_stop() function.
Affected Environment/Platform	All
Driver/Module	IGB and IXGBE Poll Mode Drivers (PMDs)

7.17 PMD does not always create rings that are properly aligned in memory

Title	PMD does not always create rings that are properly aligned in memory
Reference #	IXA00373158
Description	The NIC hardware used by the PMD requires that the RX and TX rings used must be aligned in memory on a 128-byte boundary. The memzone reservation function used inside the PMD only guarantees that the rings are aligned on a 64-byte boundary, so errors can occur if the rings are not aligned on a 128-byte boundary.
Implication	Unintended overwriting of memory can occur and PMD behavior may also be effected.
Resolution	A new rte_memzone_reserve_aligned() API has been added to allow memory reservations from hugepage memory at alignments other than 64-bytes. The PMD has been modified so that the rings are allocated using this API with minimum alignment of 128-bytes.
Affected Environment/Platform	All
Driver/Module	IGB and IXGBE Poll Mode Drivers (PMDs)

7.18 Checksum offload might not work correctly when mixing VLAN-tagged and ordinary packets

Title	Checksum offload might not work correctly when mixing VLAN-tagged and ordinary packets
Reference #	IXA00378372
Description	Incorrect handling of protocol header lengths in the PMD driver
Implication	The checksum for one of the packets may be incorrect.
Resolution/Workaround	Corrected the offset calculation.
Affected Environment/Platform	All
Driver/Module	Poll Mode Driver (PMD)

7.19 Port not found issue with Intel® 82580 Gigabit Ethernet Controller

Title	Port not found issue with Intel® 82580 Gigabit Ethernet Controller
Reference #	50
Description	After going through multiple driver unbind/bind cycles, an Intel® 82580 Ethernet Controller port may no longer be found and initialized by the DPDK.
Implication	The port will be unusable.
Resolution/Workaround	Issue was not reproducible and therefore no longer considered an issue.
Affected Environment/Platform	All
Driver/Module	1 GbE Poll Mode Driver (PMD)

7.20 Packet mbufs may be leaked from mempool if rte_eth_dev_start() function fails

Title	Packet mbufs may be leaked from mempool if rte_eth_dev_start() function fails
Reference #	IXA00373373
Description	The rte_eth_dev_start() function allocates mbufs to populate the NIC RX rings. If the start function subsequently fails, these mbufs are not freed back to the memory pool from which they came.
Implication	mbufs may be lost to the system if rte_eth_dev_start() fails and the application does not terminate.
Resolution/Workaround	mbufs are correctly deallocated if a call to rte_eth_dev_start() fails.
Affected Environment/Platform	All
Driver/Module	Poll Mode Driver (PMD)

7.21 Promiscuous mode for 82580 NICs can only be enabled after a call to rte_eth_dev_start for a port

Title	Promiscuous mode for 82580 NICs can only be enabled after a call to rte_eth_dev_start for a port
Reference #	IXA00373833
Description	For 82580-based network ports, the rte_eth_dev_start() function can overwrite the setting of the promiscuous mode for the device. Therefore, the rte_eth_promiscuous_enable() API call should be called after rte_eth_dev_start() for these devices.
Implication	Promiscuous mode can only be enabled if API calls are in a specific order.
Resolution/Workaround	The NIC now restores most of its configuration after a call to rte_eth_dev_start().
Affected Environment/Platform	All
Driver/Module	Poll Mode Driver (PMD)

7.22 Incorrect CPU socket information reported in /proc/cpuinfo can prevent the DPDK from running

Title	Incorrect CPU socket information reported in /proc/cpuinfo can prevent the Intel® DPDK from running
Reference #	63
Description	The DPDK users information supplied by the Linux kernel to determine the hardware properties of the system being used. On rare occasions, information supplied by /proc/cpuinfo does not match that reported elsewhere. In some cases, it has been observed that the CPU socket numbering given in /proc/cpuinfo is incorrect and this can prevent DPDK from operating.
Implication	The DPDK cannot run on systems where /proc/cpuinfo does not report the correct CPU socket topology.
Resolution/Workaround	CPU socket information is now read from /sys/devices/cpu/pcuN/topology
Affected Environment/Platform	All
Driver/Module	Environment Abstraction Layer (EAL)

7.23 L3FWD sample application may fail to transmit packets under extreme conditions

Title	L3FWD sample application may fail to transmit packets under extreme conditions
Reference #	IXA00372919
Description	Under very heavy load, the L3 Forwarding sample application may fail to transmit packets due to the system running out of free mbufs.
Implication	Sending and receiving data with the PMD may fail.
Resolution/ Workaround	The number of mbufs is now calculated based on application parameters.
Affected Environ- ment/Platform	All
Driver/Module	L3 Forwarding sample application

7.24 L3FWD-VF might lose CRC bytes

Title	L3FWD-VF might lose CRC bytes
Reference #	IXA00373424
Description	Currently, the CRC stripping configuration does not affect the VF driver.
Implication	Packets transmitted by the DPDK in the VM may be lacking 4 bytes (packet CRC).
Resolution/ Workaround	Set “strip_crc” to 1 in the sample applications that use the VF PMD.
Affected Environment/Platform	All
Driver/Module	IGB and IXGBE VF Poll Mode Drivers (PMDs)

7.25 32-bit DPDK sample applications fails when using more than one 1 GB hugepage

Title	32-bit Intel® DPDK sample applications fails when using more than one 1 GB hugepage
Reference #	31
Description	32-bit applications may have problems when running with multiple 1 GB pages on a 64-bit OS. This is due to the limited address space available to 32-bit processes.
Implication	32-bit processes need to use either 2 MB pages or have their memory use constrained to 1 GB if using 1 GB pages.
Resolution	EAL now limits virtual memory to 1 GB per page size.
Affected Environ- ment/Platform	64-bit systems running 32-bit Intel® DPDK with 1 GB hugepages
Driver/Module	Environment Abstraction Layer (EAL)

7.26 l2fwd fails to launch if the NIC is the Intel® 82571EB Gigabit Ethernet Controller

Title	l2fwd fails to launch if the NIC is the Intel® 82571EB Gigabit Ethernet Controller
Reference #	IXA00373340
Description	The 82571EB NIC can handle only one TX per port. The original implementation allowed for a more complex handling of multiple queues per port.
Implication	The l2fwd application fails to launch if the NIC is 82571EB.
Resolution	l2fwd now uses only one TX queue.
Affected Environment/Platform	All
Driver/Module	Sample Application

7.27 32-bit DPDK applications may fail to initialize on 64-bit OS

Title	32-bit DPDK applications may fail to initialize on 64-bit OS
Reference #	IXA00378513
Description	The EAL used a 32-bit pointer to deal with physical addresses. This could create problems when the physical address of a hugepage exceeds the 4 GB limit.
Implication	32-bit applications may not initialize on a 64-bit OS.
Resolution/Workaround	The physical address pointer is now 64-bit.
Affected Environment/Platform	32-bit applications in a 64-bit Linux* environment
Driver/Module	Environment Abstraction Layer (EAL)

7.28 Lpm issue when using prefixes > 24

Title	Lpm issue when using prefixes > 24
Reference #	IXA00378395
Description	Extended tbl8's are overwritten by multiple lpm rule entries when the depth is greater than 24.
Implication	LPM tbl8 entries removed by additional rules.
Resolution/ Workaround	Adding tbl8 entries to a valid group to avoid making the entire table invalid and subsequently overwritten.
Affected Environment/Platform	All
Driver/Module	Sample applications

7.29 IXGBE PMD hangs on port shutdown when not all packets have been sent

Title	IXGBE PMD hangs on port shutdown when not all packets have been sent
Reference #	IXA00373492
Description	When the PMD is forwarding packets, and the link goes down, and port shutdown is called, the port cannot shutdown. Instead, it hangs due to the IXGBE driver incorrectly performing the port shutdown procedure.
Implication	The port cannot shutdown and does not come back up until re-initialized.
Resolution/Workaround	The port shutdown procedure has been rewritten.
Affected Environment/Platform	All
Driver/Module	IXGBE Poll Mode Driver (PMD)

7.30 Config file change can cause build to fail

Title	Config file change can cause build to fail
Reference #	IXA00369247
Description	If a change in a config file results in some DPDK files that were needed no longer being needed, the build will fail. This is because the *.o file will still exist, and the linker will try to link it.
Implication	DPDK compilation failure
Resolution	The Makefile now provides instructions to clean out old kernel module object files.
Affected Environment/Platform	All
Driver/Module	Load balance sample application

7.31 rte_cmdline library should not be used in production code due to limited testing

Title	rte_cmdline library should not be used in production code due to limited testing
Reference #	34
Description	The rte_cmdline library provides a command line interface for use in sample applications and test applications distributed as part of DPDK. However, it is not validated to the same standard as other DPDK libraries.
Implication	It may contain bugs or errors that could cause issues in production applications.
Resolution	The rte_cmdline library is now tested correctly.
Affected Environment/Platform	All
Driver/Module	rte_cmdline

7.32 Some *_INITIALIZER macros are not compatible with C++

Title	Some *_INITIALIZER macros are not compatible with C++
Reference #	IXA00371699
Description	These macros do not work with C++ compilers, since they use the C99 method of named field initialization. The TOKEN_*_INITIALIZER macros in librte_cmdline have this problem.
Implication	C++ application using these macros will fail to compile.
Resolution/ Workaround	Macros are now compatible with C++ code.
Affected Environment/Platform	All
Driver/Module	rte_timer, rte_cmdline

7.33 No traffic through bridge when using exception_path sample application

Title	No traffic through bridge when using exception_path sample application
Reference #	IXA00168356
Description	On some systems, packets are sent from the exception_path to the tap device, but are not forwarded by the bridge.
Implication	The sample application does not work as described in its sample application guide.
Resolution/Workaround	If you cannot get packets though the bridge, it might be because IP packet filtering rules are up by default on the bridge. In that case you can disable it using the following: # for i in /proc/sys/net/bridge/bridge_nf-*; do echo 0 > $i; done
Affected Environment/Platform	Linux
Driver/Module	Exception path sample application

7.34 Segmentation Fault in testpmd after config fails

Title	Segmentation Fault in testpmd after config fails
Reference #	IXA00378638
Description	Starting testpmd with a parameter that causes port queue setup to fail, for example, set TX WTHRESH to non 0 when tx_rs_thresh is greater than 1, then doing “port start all”.
Implication	Seg fault in testpmd
Resolution/ Workaround	Testpmd now forces port reconfiguration if the initial configuration failed.
Affected Environment/Platform	All
Driver/Module	Testpmd Sample Application

7.35 Linux kernel pci_cfg_access_lock() API can be prone to deadlock

Title	Linux kernel pci_cfg_access_lock() API can be prone to deadlock
Reference #	IXA00373232
Description	The kernel APIs used for locking in the igb_uio driver can cause a deadlock in certain situations.
Implication	Unknown at this time; depends on the application.
Resolution/ Workaround	The igb_uio driver now uses the pci_cfg_access_trylock() function instead of pci_cfg_access_lock().
Affected Environment/Platform	All
Driver/Module	IGB UIO Driver

7.36 When running multi-process applications, “rte_malloc” functions cannot be used in secondary processes

Title	When running multi-process applications, “rte_malloc” functions cannot be used in secondary processes
Reference #	35
Description	The rte_malloc library provides a set of malloc-type functions that reserve memory from hugepage shared memory. Since secondary processes cannot reserve memory directly from hugepage memory, rte_malloc functions cannot be used reliably.
Implication	The librte_malloc functions, for example, rte_malloc(), rte_zmalloc() and rte_realloc() cannot be used reliably in secondary processes.
Resolution/ Workaround	In addition to re-entrancy support, the Intel® DPDK now supports the reservation of a memzone from the primary thread or secondary threads. This is achieved by putting the reservation-related control data structure of the memzone into shared memory. Since rte_malloc functions request memory directly from the memzone, the limitation for secondary threads no longer applies.
Affected Environment/Platform	All
Driver/Module	rte_malloc

7.37 Configuring maximum packet length for IGB with VLAN enabled may not take into account the length of VLAN tag

Title	Configuring maximum packet length for IGB with VLAN enabled may not take into account the length of VLAN tag
Reference #	IXA00379880
Description	For IGB, the maximum packet length configured may not include the length of the VLAN tag even if VLAN is enabled.
Implication	Packets with a VLAN tag with a size close to the maximum may be dropped.
Resolution/Workaround	NIC registers are now correctly initialized.
Affected Environment/Platform	All with IGB NICs
Driver/Module	IGB Poll Mode Driver (PMD)

7.38 Intel® I210 Ethernet controller always strips CRC of incoming packets

Title	Intel® I210 Ethernet controller always strips CRC of incoming packets
Reference #	IXA00380265
Description	The Intel® I210 Ethernet controller (NIC) removes 4 bytes from the end of the packet regardless of whether it was configured to do so or not.
Implication	Packets will be missing 4 bytes if the NIC is not configured to strip CRC.
Resolution/ Workaround	NIC registers are now correctly initialized.
Affected Environ- ment/Platform	All
Driver/Module	IGB Poll Mode Driver (PMD)

7.39 EAL can silently reserve less memory than requested

Title	EAL can silently reserve less memory than requested
Reference #	IXA00380689
Description	During application initialization, the EAL can silently reserve less memory than requested by the user through the -m application option.
Implication	The application fails to start.
Resolution	EAL will detect if this condition occurs and will give an appropriate error message describing steps to fix the problem.
Affected Environ- ment/Platform	All
Driver/Module	Environmental Abstraction Layer (EAL)

7.40 SSH connectivity with the board may be lost when starting a DPDK application

Title	SSH connectivity with the board may be lost when starting a DPDK application
Reference #	26
Description	Currently, the Intel® DPDK takes over all the NICs found on the board that are supported by the DPDK. This results in these NICs being removed from the NIC set handled by the kernel,which has the side effect of any SSH connection being terminated. See also issue #27.
Implication	Loss of network connectivity to board.
Resolution	DPDK now no longer binds ports on startup. Please refer to the Getting Started Guide for information on how to bind/unbind ports from DPDK.
Affected Environ- ment/Platform	Systems using a Intel®DPDK supported NIC for remote system access
Driver/Module	Environment Abstraction Layer (EAL)

7.41 Remote network connections lost when running autotests or sample applications

Title	Remote network connections lost when running autotests or sample applications
Reference #	27
Description	The PCI autotest and sample applications will scan for PCI devices and will remove from Linux* control those recognized by it. This may result in the loss of network connections to the system.
Implication	Loss of network connectivity to board when connected remotely.
Resolution	DPDK now no longer binds ports on startup. Please refer to the Getting Started Guide for information on how to bind/unbind ports from DPDK.
Affected Environment/Platform	Systems using a DPDK supported NIC for remote system access
Driver/Module	Sample applications

7.42 KNI may not work properly in a multi-process environment

Title	KNI may not work properly in a multi-process environment
Reference #	IXA00380475
Description	Some of the network interface operations such as, MTU change or link UP/DOWN, when executed on KNI interface, might fail in a multi-process environment, although they are normally successful in the DPDK single process environment.
Implication	Some network interface operations on KNI cannot be used in a DPDK multi-process environment.
Resolution	The ifconfig callbacks are now explicitly set in either master or secondary process.
Affected Environment/Platform	All
Driver/Module	Kernel Network Interface (KNI)

7.43 Hash library cannot be used in multi-process applications with multiple binaries

Title	Hash library cannot be used in multi-process applications with multiple binaries
Reference #	IXA00168658
Description	The hash function used by a given hash-table implementation is referenced in the code by way of a function pointer. This means that it cannot work in cases where the hash function is at a different location in the code segment in different processes, as is the case where a DPDK multi-process application uses a number of different binaries, for example, the client-server multi-process example.
Implication	The Hash library will not work if shared by multiple processes.
Resolution/Workaround	New API was added for multiprocess scenario. Please refer to DPDK Programmer's Guide for more information.
Affected Environment/Platform	All
Driver/Module	librte_hash library

7.44 Unused hugepage files are not cleared after initialization

Title	Hugepage files are not cleared after initialization
Reference #	IXA00383462
Description	EAL leaves hugepages allocated at initialization in the hugetlbfs even if they are not used.
Implication	Reserved hugepages are not freed back to the system, preventing other applications that use hugepages from running.
Resolution/Workaround	Reserved and unused hugepages are now freed back to the system.
Affected Environment/Platform	All
Driver/Module	EAL

7.45 Packet reception issues when virtualization is enabled

Title	Packet reception issues when virtualization is enabled
Reference #	IXA00369908
Description	Packets are not transmitted or received on when VT-d is enabled in the BIOS and Intel IOMMU is used. More recent kernels do not exhibit this issue.
Implication	An application requiring packet transmission or reception will not function.
Resolution/Workaround	DPDK Poll Mode Driver now has the ability to map correct physical addresses to the device structures.
Affected Environment/Platform	All
Driver/Module	Poll mode drivers

7.46 Double VLAN does not work on Intel® 40GbE Ethernet controller

Title	Double VLAN does not work on Intel® 40GbE Ethernet controller
Reference #	IXA00369908
Description	On Intel® 40 GbE Ethernet controller double VLAN does not work. This was confirmed as a Firmware issue which will be fixed in later versions of firmware.
Implication	After setting double vlan to be enabled on a port, no packets can be transmitted out on that port.
Resolution/Workaround	Resolved in latest release with firmware upgrade.
Affected Environment/Platform	All
Driver/Module	Poll mode drivers

CHAPTER

EIGHT

ABI POLICY

ABI versions are set at the time of major release labeling, and ABI may change multiple times between the last labeling and the HEAD label of the git tree without warning.

ABI versions, once released are available until such time as their deprecation has been noted here for at least one major release cycle, after it has been tagged. E.g. the ABI for DPDK 2.0 is shipped, and then the decision to remove it is made during the development of DPDK 2.1. The decision will be recorded here, shipped with the DPDK 2.1 release, and actually removed when DPDK 2.2 ships.

ABI versions may be deprecated in whole, or in part as needed by a given update.

Some ABI changes may be too significant to reasonably maintain multiple versions of. In those events ABI's may be updated without backward compatibility provided. The requirements for doing so are:

1. At least 3 acknowledgments of the need on the dpdk.org
2. A full deprecation cycle must be made to offer downstream consumers sufficient warning of the change. E.g. if dpdk 2.0 is under development when the change is proposed, a deprecation notice must be added to this file, and released with dpdk 2.0. Then the change may be incorporated for dpdk 2.1
3. The LIBABIVER variable in the makefile(s) where the ABI changes are incorporated must be incremented in parallel with the ABI changes themselves

Note that the above process for ABI deprecation should not be undertaken lightly. ABI stability is extremely important for downstream consumers of the DPDK, especially when distributed in shared object form. Every effort should be made to preserve ABI whenever possible. For instance, reorganizing public structure field for aesthetic or readability purposes should be avoided as it will cause ABI breakage. Only significant (e.g. performance) reasons should be seen as cause to alter ABI.

8.1 Examples of Deprecation Notices

- The Macro #RTE_FOO is deprecated and will be removed with version 2.0, to be replaced with the inline function rte_bar()
- The function rte_mbuf_grok has been updated to include new parameter in version 2.0. Backwards compatibility will be maintained for this function until the release of version 2.1

- The members struct foo have been reorganized in release 2.0. Existing binary applications will have backwards compatibility in release 2.0, while newly built binaries will need to reference new structure variant struct foo2. Compatibility will be removed in release 2.2, and all applications will require updating and rebuilding to the new structure at that time, which will be renamed to the original struct foo.
- Significant ABI changes are planned for the librte_dostuff library. The upcoming release 2.0 will not contain these changes, but release 2.1 will, and no backwards compatibility is planned due to the invasive nature of these changes. Binaries using this library built prior to version 2.1 will require updating and recompilation.

8.2 Deprecation Notices

CHAPTER

NINE

FREQUENTLY ASKED QUESTIONS (FAQ)

9.1 When running the test application, I get "EAL: map_all_hugepages(): open failed: Permission denied Cannot init memory"?

This is most likely due to the test application not being run with sudo to promote the user to a superuser. Alternatively, applications can also be run as regular user. For more information, please refer to *DPDK Getting Started Guide*.

9.2 If I want to change the number of TLB Hugepages allocated, how do I remove the original pages allocated?

The number of pages allocated can be seen by executing the cat /proc/meminfo|grep Huge command. Once all the pages are mmapped by an application, they stay that way. If you start a test application with less than the maximum, then you have free pages. When you stop and restart the test application, it looks to see if the pages are available in the /dev/huge directory and mmaps them. If you look in the directory, you will see n number of 2M pages files. If you specified 1024, you will see 1024 files. These are then placed in memory segments to get contiguous memory.

If you need to change the number of pages, it is easier to first remove the pages. The tools/setup.sh script provides an option to do this. See the "Quick Start Setup Script" section in the *DPDK Getting Started Guide* for more information.

9.3 If I execute "l2fwd -c f -m 64 –n 3 – -p 3", I get the following output, indicating that there are no socket 0 hugepages to allocate the mbuf and ring structures to?

I have set up a total of 1024 Hugepages (that is, allocated 512 2M pages to each NUMA node).

The -m command line parameter does not guarantee that huge pages will be reserved on specific sockets. Therefore, allocated huge pages may not be on socket 0. To request memory to be reserved on a specific socket, please use the –socket-mem command-line parameter instead of -m.

9.4 I am running a 32-bit DPDK application on a NUMA system, and sometimes the application initializes fine but cannot allocate memory. Why is that happening?

32-bit applications have limitations in terms of how much virtual memory is available, hence the number of hugepages they are able to allocate is also limited (1 GB per page size). If your system has a lot (>1 GB per page size) of hugepage memory, not all of it will be allocated. Due to hugepages typically being allocated on a local NUMA node, the hugepages allocation the application gets during the initialization depends on which NUMA node it is running on (the EAL does not affinitize cores until much later in the initialization process). Sometimes, the Linux OS runs the DPDK application on a core that is located on a different NUMA node from DPDK master core and therefore all the hugepages are allocated on the wrong socket.

To avoid this scenario, either lower the amount of hugepage memory available to 1 GB per page size (or less), or run the application with taskset affinitizing the application to a would-be master core. For example, if your EAL coremask is 0xff0, the master core will usually be the first core in the coremask (0x10); this is what you have to supply to taskset, for example, taskset 0x10 ./l2fwd -c 0xff0 -n 2. In this way, the hugepages have a greater chance of being allocated to the correct socket. Additionally, a –socket-mem option could be used to ensure the availability of memory for each socket, so that if hugepages were allocated on the wrong socket, the application simply will not start.

9.5 On application startup, there is a lot of EAL information printed. Is there any way to reduce this?

Yes, each EAL has a configuration file that is located in the /config directory. Within each configuration file, you will find CONFIG_RTE_LOG_LEVEL=8. You can change this to a lower value, such as 6 to reduce this printout of debug information. The following is a list of LOG levels that can be found in the rte_log.h file. You must remove, then rebuild, the EAL directory for the change to become effective as the configuration file creates the rte_config.h file in the EAL directory.

```
#define RTE_LOG_EMERG 1U    /* System is unusable. */
#define RTE_LOG_ALERT 2U    /* Action must be taken immediately. */
#define RTE_LOG_CRIT 3U     /* Critical conditions. */
#define RTE_LOG_ERR 4U      /* Error conditions. */
#define RTE_LOG_WARNING 5U  /* Warning conditions. */
#define RTE_LOG_NOTICE 6U   /* Normal but significant condition. */
#define RTE_LOG_INFO 7U     /* Informational. */
#define RTE_LOG_DEBUG 8U    /* Debug-level messages. */
```

9.6 How can I tune my network application to achieve lower latency?

Traditionally, there is a trade-off between throughput and latency. An application can be tuned to achieve a high throughput, but the end-to-end latency of an average packet typically increases as a result. Similarly, the application can be tuned to have, on average, a low end-to-end latency at the cost of lower throughput.

To achieve higher throughput, the DPDK attempts to aggregate the cost of processing each packet individually by processing packets in bursts. Using the testpmd application as an example, the "burst" size can be set on the command line to a value of 16 (also the default value). This allows the application to request 16 packets at a time from the PMD. The testpmd application then immediately attempts to transmit all the packets that were received, in this case, all 16 packets. The packets are not transmitted until the tail pointer is updated on the corresponding TX queue of the network port. This behavior is desirable when tuning for high throughput because the cost of tail pointer updates to both the RX and TX queues can be spread across 16 packets, effectively hiding the relatively slow MMIO cost of writing to the PCIe* device.

However, this is not very desirable when tuning for low latency, because the first packet that was received must also wait for the other 15 packets to be received. It cannot be transmitted until the other 15 packets have also been processed because the NIC will not know to transmit the packets until the TX tail pointer has been updated, which is not done until all 16 packets have been processed for transmission.

To consistently achieve low latency even under heavy system load, the application developer should avoid processing packets in bunches. The testpmd application can be configured from the command line to use a burst value of 1. This allows a single packet to be processed at a time, providing lower latency, but with the added cost of lower throughput.

9.7 Without NUMA enabled, my network throughput is low, why?

I have a dual Intel® Xeon® E5645 processors @2.40 GHz with four Intel® 82599 10 Gigabit Ethernet NICs. Using eight logical cores on each processor with RSS set to distribute network load from two 10 GbE interfaces to the cores on each processor.

Without NUMA enabled, memory is allocated from both sockets, since memory is interleaved. Therefore, each 64B chunk is interleaved across both memory domains.

The first 64B chunk is mapped to node 0, the second 64B chunk is mapped to node 1, the third to node 0, the fourth to node 1. If you allocated 256B, you would get memory that looks like this:

```
256B buffer
Offset 0x00 - Node 0
Offset 0x40 - Node 1
Offset 0x80 - Node 0
Offset 0xc0 - Node 1
```

Therefore, packet buffers and descriptor rings are allocated from both memory domains, thus incurring QPI bandwidth accessing the other memory and much higher latency. For best performance with NUMA disabled, only one socket should be populated.

9.8 I am getting errors about not being able to open files. Why?

As the DPDK operates, it opens a lot of files, which can result in reaching the open files limits, which is set using the ulimit command or in the limits.conf file. This is especially true when using a large number (>512) of 2 MB huge pages. Please increase the open file limit if your application is not able to open files. This can be done either by issuing a ulimit command or editing the limits.conf file. Please consult Linux* manpages for usage information.

9.9 Does my kernel require patching to run the DPDK?

Any kernel greater than version 2.6.33 can be used without any patches applied. The following kernels may require patches to provide hugepage support:

- kernel version 2.6.32 requires the following patches applied:
  - addhugepage support to pagemap
  - fix hugepage memory leak
  - add nodemask arg to huge page alloc

    (not mandatory, but recommended on a NUMA system to support per-NUMA node hugepages allocation)
- kernel version 2.6.31, requires the following patches applied:
  - fix hugepage memory leak
  - add hugepage support to pagemap
  - add uio name attributes and port regions
  - add nodemask arg to huge page alloc

    (not mandatory, but recommended on a NUMA system to support per-NUMA node hugepages allocation)

Note: Blue text in the lists above are direct links to the patch downloads.

9.10 VF driver for IXGBE devices cannot be initialized.

Some versions of Linux* IXGBE driver do not assign a random MAC address to VF devices at initialization. In this case, this has to be done manually on the VM host, using the following command:

```
ip link set <interface> vf <VF function> mac <MAC address>
```

where <interface> being the interface providing the virtual functions for example, eth0, <VF function> being the virtual function number, for example 0, and <MAC address> being the desired MAC address.

9.11 Is it safe to add an entry to the hash table while running?

Currently the table implementation is not a thread safe implementation and assumes that locking between threads and processes is handled by the user's application. This is likely to be supported in future releases.

9.12 What is the purpose of setting iommu=pt?

DPDK uses a 1:1 mapping and does not support IOMMU. IOMMU allows for simpler VM physical address translation. The second role of IOMMU is to allow protection from unwanted

memory access by an unsafe device that has DMA privileges. Unfortunately, the protection comes with an extremely high performance cost for high speed NICs.

iommu=pt disables IOMMU support for the hypervisor.

9.13 When trying to send packets from an application to itself, meaning smac==dmac, using Intel(R) 82599 VF packets are lost.

Check on register LLE(PFVMTXSSW[n]), which allows an individual pool to send traffic and have it looped back to itself.

9.14 Can I split packet RX to use DPDK and have an application's higher order functions continue using Linux* pthread?

The DPDK's lcore threads are Linux* pthreads bound onto specific cores. Configure the DPDK to do work on the same cores and run the application's other work on other cores using the DPDK's "coremask" setting to specify which cores it should launch itself on.

9.15 Is it possible to exchange data between DPDK processes and regular userspace processes via some shared memory or IPC mechanism?

Yes - DPDK processes are regular Linux/BSD processes, and can use all OS provided IPC mechanisms.

9.16 Can the multiple queues in Intel(R) I350 be used with DPDK?

I350 has RSS support and 8 queue pairs can be used in RSS mode. It should work with multi-queue DPDK applications using RSS.

9.17 How can hugepage-backed memory be shared among multiple processes?

See the Primary and Secondary examples in the multi-process sample application.