



NVIDIA MLNX_EN Documentation

v5.8-5.1.1.2 LTS

Table of Contents

1	Release Notes.....	5
1.1	Release Notes Update History.....	5
1.2	Supported NIC Speeds	5
1.3	Embedded Components	6
1.4	General Support.....	7
1.4.1	Upgrade/Downgrade Matrix	15
1.4.2	Supported NIC Firmware Versions	15
1.5	Changes and New Features.....	16
1.5.1	New Features.....	16
1.5.2	Unsupported Functionalities/Features/NICs.....	17
1.6	Bug Fixes in This Version.....	17
1.7	Known Issues.....	18
2	User Manual.....	48
2.1	Introduction.....	48
2.1.1	Package Contents	49
2.1.2	Module Parameters	50
2.1.3	Devlink Parameters	50
2.2	Installation.....	51
2.2.1	Software Dependencies	51
2.2.2	Downloading the Drivers.....	51
2.2.3	Installing MLNX_EN	52
2.2.4	Uninstall.....	58
2.2.5	Updating Firmware After Installation.....	58
2.2.6	Ethernet Driver Usage and Configuration	59
2.2.7	Performance Tuning	61
2.3	Features Overview and Configuration	61
2.3.1	Ethernet Network.....	61
2.3.2	Virtualization	108
2.3.3	Resiliency	124
2.3.4	Docker Containers	126
2.3.5	Fast Driver Unload	127
2.3.6	OVS Offload Using ASAP ² Direct	127

2.4	Troubleshooting	166
2.4.1	General Issues	167
2.4.2	Ethernet Related Issues	168
2.4.3	Installation Related Issues	169
2.4.4	Performance Related Issues	171
2.4.5	SR-IOV Related Issues	171
2.4.6	OVS Offload Using ASAP2 Direct Related Issues	172
2.5	Common Abbreviations and Related Documents	172
3	Documentation History.....	175
3.1	Release Notes History	175
3.1.1	Release Notes Change Log History	175
3.1.2	Bug Fixes History.....	201
3.2	User Manual Revision History.....	238
4	Legal Notices and 3rd Party Licenses	239

Overview

NVIDIA offers a robust and full set of protocol software and driver for Linux with the ConnectX® EN family cards. Designed to provide a high performance support for Enhanced Ethernet with fabric consolidation over TCP/IP based LAN applications. The driver and software in conjunction with the industry's leading ConnectX family of cards achieve full line rate, full duplex of up to 400GbE performance per port.

Further information on this product can be found in the following MLNX_EN documents:

- [Release Notes](#)
- [User Manual](#)

Software Download

Please visit nvidia.com/en-us/networking → Products → Software → Ethernet Drivers → [NVIDIA EN for Linux](#)

Document Revision History

For the list of changes made to the User Manual, refer to [User Manual Revision History](#).

For the list of changes made to the Release Notes, refer to [Release Notes Revision History](#).

1 Release Notes

This is a long-term support (LTS) release. LTS is the practice of maintaining a software product for an extended period of time (up to three years) to help increase product stability. LTS releases include bug fixes and security patches.

This LTS version should be used by customers who would like to utilize ConnectX-4 adapter cards and above and keep using stable 5.8-x deployment while getting the following:

- Critical bug fixes
- Support for new major operating systems

For other use cases, it is recommended to use the latest 5.x version.

1.1 Release Notes Update History

Version	Date	Description
5.8-5.1.1.2	July 04, 2024	Fifth update of the 5.8 LTS branch.

As of MLNX_EN version 5.1-1.0.4.0, the following are no longer supported.

- ConnectX-3
- ConnectX-3 Pro
- Connect-IB
- RDMA experimental verbs libraries (mlnx_lib)

To utilize the above devices/libraries, refer to version 4.9 long-term support (LTS).

Release Notes contain the following sections:

- [General Support](#)
- [Changes and New Features](#)
- [Bug Fixes in This Version](#)
- [Known Issues](#)

1.2 Supported NIC Speeds

The Linux Driver operates across all NVIDIA network adapter solutions supporting the following uplinks to servers:

Uplink/Adapter Card	Driver Name	Uplink Speed
BlueField-2	mlx5	<ul style="list-style-type: none">• InfiniBand: SDR, FDR, EDR, HDR• Ethernet: 1GbE, 10GbE, 25GbE, 40GbE, 50GbE², 100GbE²
BlueField		<ul style="list-style-type: none">• InfiniBand: SDR, QDR, FDR, FDR10, EDR• Ethernet: 1GbE, 10GbE, 25GbE, 40GbE, 50GbE, 100GbE

Uplink/Adapter Card	Driver Name	Uplink Speed
ConnectX-7		<ul style="list-style-type: none"> InfiniBand: EDR, HDR100, HDR, NDR200, NDR Ethernet: 1GbE, 10GbE, 25GbE, 40GbE, 50GbE², 100GbE², 200GbE³, 400GbE
ConnectX-6 Lx		<ul style="list-style-type: none"> Ethernet: 1GbE, 10GbE, 25GbE, 40GbE, 50GbE²
ConnectX-6 Dx		<ul style="list-style-type: none"> Ethernet: 10GbE, 25GbE, 40GbE, 50GbE², 100GbE², 200GbE²
ConnectX-6		<ul style="list-style-type: none"> InfiniBand: SDR, FDR, EDR, HDR Ethernet: 10GbE, 25GbE, 40GbE, 50GbE², 100GbE², 200GbE²
ConnectX-5/ConnectX-5 Ex		<ul style="list-style-type: none"> InfiniBand: SDR, QDR, FDR, FDR10, EDR Ethernet: 1GbE, 10GbE, 25GbE, 40GbE, 50GbE, 100GbE
ConnectX-4 Lx		<ul style="list-style-type: none"> Ethernet: 1GbE, 10GbE, 25GbE, 40GbE, 50GbE
ConnectX-4		<ul style="list-style-type: none"> InfiniBand: SDR, QDR, FDR, FDR10, EDR Ethernet: 1GbE, 10GbE, 25GbE, 40GbE, 50GbE, 56GbE¹, 100GbE

1. 56GbE is an NVIDIA proprietary link speed and can be achieved while connecting an NVIDIA adapter card to NVIDIA SX10XX switch series or when connecting an NVIDIA adapter card to another NVIDIA adapter card.
2. Speed that supports both NRZ and PAM4 modes in Force mode and Auto-Negotiation mode.
3. Speed that supports PAM4 mode only.

1.3 Embedded Components

Package	Revision	Licenses
clusterkit	1.8.428-1.58511	BSD
dapl	2.1.10.1.mlnx-OFED.4.9.0.1.5.58511	Dual GPL/BSD/CPL
dpcp	1.1.37-1.58511	BSD-3-Clause
dump_pr	1.0-5.13.0.MLNX20221016.gac314ef.58511	GPLv2 or BSD
hcoll	4.8.3220-1.58511	Proprietary
ibdump	6.0.0-1.58511	BSD2+GPL2
ibsim	0.10-1.58511	GPLv2 or BSD
ibutils2	2.1.1-0.156.MLNX20221016.g4aceb16.58511	Mellanox Confidential and Proprietary
iser	5.8-OFED.5.8.5.1.1.1	GPLv2
isert	5.8-OFED.5.8.5.1.1.1	GPLv2
kernel-mft	4.22.1-417	Dual BSD/GPL
knem	1.1.4.90mlnx3-OFED.23.10.0.2.1.1	BSD and GPLv2
libvma	9.7.2-1	GPLv2 or BSD
libxlio	2.0.7-1	GPLv2 or BSD
mlnx-en	5.8-5.1.1.0.g4370d14	GPLv2

Package	Revision	Licenses
mlnx-ethtool	5.18-1.58511	GPL
mlnx-iproute2	5.19.0-1.58511	GPL
mlnx-nfsrdma	5.8-OFED.5.8.5.1.1.1	GPLv2
mlnx-nvme	5.8-OFED.5.8.5.1.1.1	GPLv2
mlnx-ofa_kernel	5.8-OFED.5.8.5.1.1.1	GPLv2
mlnx-tools	5.8.0-1.lts.58511	GPLv2
mlx-steering-dump	1.0.0-0.58511	GPLv2
mpi-selector	1.0.3-1.58511	BSD
mpitests	3.2.20-de56b6b.58511	BSD
mstflint	4.16.1-2.58511	GPL/BSD
multiperf	3.0-3.0.58511	BSD 3-Clause, GPL v2 or later
ofed-docs	5.8-OFED.5.8.5.1.1	GPL/BSD
ofed-scripts	5.8-OFED.5.8.5.1.1	GPL/BSD
openmpi	4.1.5a1-1.58511	BSD
opensm	5.13.0.MLNX20221016.10d3954-0.1.58511	GPLv2 or BSD
openvswitch	2.17.2-1.58511	ASL 2.0 and LGPLv2+ and SISSL
perftest	4.5-0.18.gfcddfe0.58511	BSD 3-Clause, GPL v2 or later
rdma-core	58mlnx43-1.58511	GPLv2 or BSD
rshim	2.0.19-0.gbf7f1f2	GPLv2
sharp	3.1.1.MLNX20221122.c93d7550-1.58511	Proprietary
sockperf	3.10-0.git5ebd327da983.58511	BSD
srp	5.8-OFED.5.8.5.1.1.1	GPLv2
ucx	1.14.0-1.58511	BSD
xpmem	2.7.3-1.58511	GPLv2 and LGPLv2.1

1.4 General Support

Supported Operating Systems

Operating System	Architecture	Default Kernel Version (Primary)/ Tested with Kernel Version (Community)	OS Support Model	ASAP ² OVS-Kernel SR-IOV	ASAP ² OVS-DPDK SR-IOV	NFS on RDM A	NVMe	GPU Direct Storage (GDS)	UCX-CUDA Version	NEO-Host
ALIOS7.2	AArch64	4.19.48-006.ali4000.alios7.aarch64	Primary	✗	✗	✗	✗	✗	✗	✗
Alma 8.5	x86_64	4.18.0-348.12.2.EL8_5.X86_64	Community	✗	✗	✗	✗	✗	✗	✗
Anolis OS 8.4	AArch64	4.18.0-348.2.1.AN8_4.AARCH64	Community	✗	✗	✗	✗	✗	✗	✗
	x86_64	4.18.0-305.AN8.X86_64	Community	✗	✗	✗	✗	✗	✗	✗
BCLINUX21.1 OSP2	AArch64	4.19.90-2107.6.0.0098.oel1.bclinux.aarch64	Primary	✗	✗	✗	✗	✗	✗	✗
BCLINUX8.1	AArch64		Community	✗	✗	✗	✗	✗	✗	✗
	x86_64	4.19.0-193.1.3.el8.bclinux.x86_64	Community	✗	✗	✗	✗	✗	✗	✗
BCLINUX8.2	x86_64	4.19.0-240.23.11.el8_2.bclinux.x86_64	Primary	✗	✗	✗	✗	✗	✗	✗
CentOS Stream v8	AArch64	4.18.0-529.el8.aarch64	Community	✗	✗	✗	✗	✗	✗	✗
	x86_64	4.18.0-529.el8.x86_64	Community	✗	✗	✗	✗	✗	✗	✗
CentOS Stream v9	AArch64	5.14.0-391.el9.aarch64	Community	✗	✗	✗	✗	✗	✗	✗
	x86_64	5.14.0-391.el9.x86_64	Community	✗	✗	✗	✗	✗	✗	✗
CTYUNOS2.0	AArch64	4.19.90-2102.2.0.0062.clt2.aarch64	Primary	✗	✗	✗	✗	✗	✗	✗
	x86_64	4.19.90-2102.2.0.0062.clt2.x86_64	Primary	✗	✗	✗	✗	✗	✗	✗
Debian10.8	AArch64	4.19.0-14-arm64	Primary	✗	✗	✗	✗	✗	✗	✗
	x86_64	4.19.0-14-amd64	Primary	✗	✗	✗	✗	✗	✗	✗
Debian10.9	x86_64	4.19.0-16-amd64	Primary	✓	✗	✗	✓	✗	✗	✗
Debian10.13	AArch64	4.19.0-21-arm64	Primary	✓	✗	✗	✓	✗	✗	✗

	x86_64	4.19.0-21-amd64	Primary	✓	✗	✗	✓	✗	✗	✗
Debian11.3	AArch64	5.10.0-13-arm64	Primary	✓	✗	✗	✓	✗	✗	✗
	x86_64	5.10.0-13-amd64	Primary	✓	✗	✗	✓	✗	✗	✗
Debian9.13	AArch64	4.9.0-13-arm64	Primary	✗	✗	✗	✗	✗	✗	✗
	x86_64	4.9.0-13-amd64	Primary	✗	✗	✗	✗	✗	✗	✗
EulerOS2.0sp8	AArch64	4.19.36-vhulk1907.1.0.h748.eulerosv2r8.aarch64	Community	✗	✗	✗	✗	✗	✗	✗
EulerOS2.0sp10	AArch64	4.19.90-vhulk2110.1.0.h860.eulerosv2r10.aarch64	Primary	✗	✗	✗	✗	✗	✗	✗
	x86_64	4.18.0-147.5.2.4.h694.eulerosv2r10.x86_64	Primary	✗	✗	✗	✗	✗	✗	✗
EulerOS2.0sp11	AArch64	5.10.0-60.18.0.50.h323.eulerosv2r11.aarch64	Primary	✗	✗	✗	✗	✗	✗	✗
	x86_64	5.10.0-60.18.0.50.h323.eulerosv2r11.x86_64	Primary	✗	✗	✗	✗	✗	✗	✗
EulerOS2.0sp9	AArch64	4.19.90-vhulk2006.2.0.h171.eulerosv2r9.aarch64	Primary	✗	✗	✗	✗	✗	✗	✗
	x86_64	4.18.0-147.5.1.0.h269.eulerosv2r9.x86_64	Primary	✗	✗	✗	✗	✗	✗	✗
Fedora 35	x86_64	5.16.8-200.fc35.x86_64	Community	✗	✗	✗	✗	✗	✗	✗
KYLIN10	AArch64	4.19.90-17.ky10.aarch64	Primary	✗	✗	✗	✗	✓	✗	✗
	x86_64	4.19.90-17.ky10.x86_64	Primary	✗	✗	✗	✗	✓	✗	✗
KYLIN10SP2	AArch64	4.19.90-24.4.v2101.ky10.aarch64	Primary	✗	✗	✗	✗	✗	✗	✗
	x86_64	4.19.90-24.4.v2101.ky10.x86_64	Primary	✓	✗	✗	✗	✗	✗	✓
KYLIN10SP3	AArch64	4.19.90-52.15.v2207.ky10.aarch64	Primary	✗	✗	✗	✗	✗	✗	✗
	x86_64	4.19.90-52.15.v2207.ky10.x86_64	Primary	✓	✗	✗	✗	✗	✗	✓
Oracle Linux 7.9	x86_64	5.4.17-2011.6.2.el7uek.x86_64	Primary	✓	✗	✗	✗	✗	✗	✗
Oracle Linux 8.4	x86_64	5.4.17-2102.201.3.el8uek.x86_64	Primary	✓	✗	✗	✗	✗	✗	✗

Oracle Linux 8.6	x86_64	5.4.17-2136.307.3.1.el8uek.x86_64	Primary	✓	✗	✗	✗	✗	✗	✗	✗
Oracle Linux 9.0	x86_64	5.15.0-0.30.19.el9uek.x86_64	Primary	✗	✗	✗	✗	✗	✗	✗	✗
OpenSUSE 15.3	AArch64	-	Community	✗	✗	✗	✗	✗	✗	✗	✗
	ppc64le	-	Community	✗	✗	✗	✗	✗	✗	✗	✗
	x86_64	5.3.18-150300.59.43-DEFAULT	Community	✗	✗	✗	✗	✗	✗	✗	✗
OPENEULER2 0.03SP3	AArch64	4.19.90-2112.8.0.0131.oe1.aarch64	Primary	✗	✗	✗	✗	✗	✗	✗	✗
	x86_64	4.19.90-2112.8.0.0131.oe1.x86_64	Primary	✗	✗	✗	✗	✗	✗	✗	✗
OPENEULER2 2.03	AArch64	5.10.0-60.18.0.50.oe2203.aarch64	Primary	✗	✗	✗	✗	✗	✗	✗	✗
	x86_64	5.10.0-60.18.0.50.oe2203.x86_64	Primary	✗	✗	✗	✗	✗	✗	✗	✗
Photon OS 3.0	x86_64	4.19.225-3.ph3	Community	✗	✗	✗	✗	✗	✗	✗	✗
RHEL/CentOS7.2	x86_64	3.10.0-327.el7.x86_64	Primary	✗	✗	✗	✓	✗	11.7	✗	✗
RHEL/CentOS7.4	ppc64	3.10.0-693.el7.ppc64	Primary	✓	✓	✗	✓	✗	✗	✗	✓
	ppc64le	3.10.0-693.el7.ppc64le	Primary	✓	✓	✗	✓	✗	✗	✗	✓
	x86_64	3.10.0-693.el7.x86_64	Primary	✓	✓	✗	✓	✗	11.7	✓	✓
RHEL/CentOS7.6	ppc64	3.10.0-957.el7.ppc64	Primary	✓	✓	✗	✓	✗	✗	✗	✓
	ppc64le	3.10.0-957.el7.ppc64le	Primary	✓	✓	✗	✓	✗	✗	✗	✓
	x86_64	3.10.0-957.el7.x86_64	Primary	✓	✓	✓	✓	✗	11.7	✓	✓
RHEL/CentOS7.6al ternate	aarch64	4.14.0-115.el7a.aarch64	Community	✓	✓	✗	✓	✗	✗	✗	✓
	ppc64le	4.14.0-115.el7a.ppc64le	Community	✓	✓	✗	✓	✗	✗	✗	✓
RHEL/CentOS7.7	x86_64	3.10.0-1062.el7.x86_64	Primary	✓	✓	✗	✗	✗	11.7	✓	✓
RHEL/CentOS7.8	ppc64	3.10.0-1127.el7.ppc64	Primary	✓	✓	✗	✓	✗	✗	✗	✓
	ppc64le	3.10.0-1127.el7.ppc64le	Primary	✓	✓	✗	✓	✗	✗	✗	✓

	x86_64	3.10.0-1127.el7.x86_64	Primary	✓	✓	✓	✓	✗	11.7	✓
RHEL/ CentOS7.9	ppc64	3.10.0-1160.el7.ppc64	Primary	✓	✓	✗	✓	✗	✗	✓
	ppc64le	3.10.0-1160.el7.ppc64le	Primary	✓	✓	✗	✓	✗	✗	✓
	x86_64	3.10.0-1160.el7.x86_64	Primary	✓	✓	✓	✓	✗	11.7	✓
RHEL/ CentOS8.0	AArch64	4.18.0-80.el8.aarch64	Primary	✓	✓	✗	✗	✓	11.7	✓
	ppc64le	4.18.0-80.el8.ppc64le	Primary	✓	✓	✗	✗	✓	✗	✓
	x86_64	4.18.0-80.el8.x86_64	Primary	✓	✓	✗	✗	✓	11.7	✓
RHEL/ CentOS8.1	AArch64	4.18.0-147.el8.aarch64	Primary	✓	✓	✗	✗	✓	11.7	✓
	ppc64le	4.18.0-147.el8.ppc64le	Primary	✓	✓	✗	✗	✓	✗	✓
	x86_64	4.18.0-147.el8.x86_64	Primary	✓	✓	✗	✗	✓	11.7	✓
RHEL/ CentOS8.2	AArch64	4.18.0-193.el8.aarch64	Primary	✓	✓	✗	✓	✓	11.7	✓
	ppc64le	4.18.0-193.el8.ppc64le	Primary	✓	✓	✗	✓	✓	✗	✓
	x86_64	4.18.0-193.el8.x86_64	Primary	✓	✓	✓	✓	✓	11.7	✓
RHEL/ CentOS8.3	AArch64	4.18.0-240.el8.aarch64	Primary	✓	✓	✗	✗	✓	11.7	✓
	ppc64le	4.18.0-240.el8.ppc64le	Primary	✓	✓	✗	✗	✓	✗	✓
	x86_64	4.18.0-240.el8.x86_64	Primary	✓	✓	✗	✗	✓	11.7	✓
RHEL/ CentOS8.4	AArch64	4.18.0-305.el8.aarch64	Primary	✓	✓	✗	✓	✓	11.7	✓
	ppc64le	4.18.0-305.el8.ppc64le	Primary	✓	✓	✗	✓	✓	✗	✓
	x86_64	4.18.0-305.el8.x86_64	Primary	✓	✓	✓	✓	✓	11.7	✓
RHEL/ CentOS/ Rocky8.5	AArch64	4.18.0-348.el8.aarch64	Primary	✓	✓	✗	✓	✓	11.7	✓
	ppc64le	4.18.0-348.el8.ppc64le	Primary	✓	✓	✗	✓	✓	✗	✓
	x86_64	4.18.0-348.el8.x86_64	Primary	✓	✓	✓	✓	✓	11.7	✓
RHEL/ Rocky8.6	AArch64	4.18.0-372.41.1.el8_6.aarch64	Primary	✓	✗	✗	✓	✓	11.7	✓

	ppc64le	4.18.0-372.41.1.el8_6.ppc64le	Primary	✓	✗	✗	✓	✓	✗	✓
	x86_64	4.18.0-372.41.1.el8_6.x86_64	Primary	✓	✗	✓	✓	✓	11.7	✓
RHEL/ Rocky8.7	AArch64	4.18.0-425.14.1.el8_7.aarch64	Primary	✓	✗	✗	✓	✓	11.7	✓
	ppc64le	4.18.0-425.14.1.el8_7.ppc64le	Primary	✓	✗	✗	✓	✓	✗	✓
	x86_64	4.18.0-425.14.1.el8_7.x86_64	Primary	✓	✗	✓	✓	✓	11.7	✓
RHEL/ Rocky8.8	AArch64	4.18.0-477.10.1.el8_8.aarch64	Primary	✓	✗	✗	✓	✓	11.7	✓
	ppc64le	4.18.0-477.10.1.el8_8.ppc64le	Primary	✓	✗	✗	✓	✓	✗	✓
	x86_64	4.18.0-477.10.1.el8_8.x86_64	Primary	✓	✗	✓	✓	✓	11.7	✓
RHEL/ Rocky8.9	AArch64	4.18.0-513.5.1.el8_9.aarch64	Primary	✓	✗	✗	✓	✓	11.7	✓
	ppc64le	4.18.0-513.5.1.el8_9.ppc64le	Primary	✓	✗	✗	✓	✓	✗	✓
	x86_64	4.18.0-513.5.1.el8_9.x86_64	Primary	✓	✗	✓	✓	✓	11.7	✓
RHEL/ Rocky9.0	AArch64	5.14.0-70.46.1.el9_0.aarch64	Primary	✓	✗	✗	✓	✓	11.7	✗
	ppc64le	5.14.0-70.46.1.el9_0.ppc64le	Primary	✓	✗	✗	✓	✓	✗	✗
	x86_64	5.14.0-70.46.1.el9_0.x86_64	Primary	✓	✗	✓	✓	✓	11.7	✗
RHEL/ Rocky9.1	AArch64	5.14.0-162.19.1.el9_1.aarch64	Primary	✓	✗	✗	✓	✓	11.7	✗
	ppc64le	5.14.0-162.19.1.el9_1.ppc64le	Primary	✓	✗	✗	✓	✓	✗	✗
	x86_64	5.14.0-162.19.1.el9_1.x86_64	Primary	✓	✗	✓	✓	✓	11.7	✗
RHEL/ Rocky9.2	AArch64	5.14.0-284.11.1.el9_2.aarch64	Primary	✓	✗	✗	✓	✓	11.7	✗
	ppc64le	5.14.0-284.11.1.el9_2.ppc64le	Primary	✓	✗	✗	✓	✓	✗	✗
	x86_64	5.14.0-284.11.1.el9_2.x86_64	Primary	✓	✗	✓	✓	✓	11.7	✗
RHEL/ Rocky9.3	AArch64	5.14.0-362.8.1.el9_3.aarch64	Primary	✓	✗	✗	✓	✓	11.7	✗
	ppc64le	5.14.0-362.8.1.el9_3.ppc64le	Primary	✓	✗	✗	✓	✓	✗	✗
	x86_64	5.14.0-362.8.1.el9_3.x86_64	Primary	✓	✗	✓	✓	✓	11.7	✗

SLES12SP2	x86_64	4.4.21-69-default	Community	✗	✗	✗	✗	✗	✗	✗
SLES12SP3	ppc64le	4.4.73-5-default	Community	✗	✗	✗	✗	✗	✗	✓
	x86_64	4.4.73-5-default	Community	✗	✗	✗	✗	✗	✗	✓
SLES12SP4	AArch64	4.12.14-94.41-default	Community	✓	✗	✗	✓	✗	✗	✓
	ppc64le	4.12.14-94.41-default	Community	✓	✗	✗	✓	✗	✗	✓
	x86_64	4.12.14-94.41-default	Community	✓	✗	✗	✓	✗	✗	✓
SLES12SP5	AArch64	4.12.14-120-default	Primary	✓	✗	✓	✓	✗	✗	✓
	ppc64le	4.12.14-120-default	Primary	✓	✗	✓	✓	✗	✗	✓
	x86_64	4.12.14-120-default	Primary	✓	✗	✓	✓	✗	✗	✓
SLES15SP2	AArch64	5.3.18-22-default	Primary	✓	✓	✓	✓	✗	✗	✓
	ppc64le	5.3.18-22-default	Primary	✓	✓	✓	✓	✗	✗	✓
	x86_64	5.3.18-22-default	Primary	✓	✓	✓	✓	✗	✗	✓
SLES15SP3	AArch64	5.3.18-57-default	Primary	✓	✗	✓	✓	✗	✗	✓
	ppc64le	5.3.18-57-default	Primary	✓	✗	✓	✓	✗	✗	✓
	x86_64	5.3.18-57-default	Primary	✓	✗	✓	✓	✗	✗	✓
SLES15SP4	AArch64	5.14.21-150400.22-default	Primary	✓	✗	✓	✓	✗	✗	✗
	ppc64le	5.14.21-150400.22-default	Primary	✓	✗	✓	✓	✗	✗	✗
	x86_64	5.14.21-150400.22-default	Primary	✓	✗	✓	✓	✗	✗	✗
SLES15SP5	AArch64	5.14.21-150500.53-default	Primary	✓	✗	✓	✓	✗	✗	✗
	ppc64le	5.14.21-150500.53-default	Primary	✓	✗	✓	✓	✗	✗	✗
	x86_64	5.14.21-150500.53-default	Primary	✓	✗	✓	✓	✗	✗	✗
Ubuntu16.04	ppc64le	4.4.0-21-generic	Community	✗	✗	✗	✗	✗	✗	✗
	x86_64	4.4.0-21-generic	Community	✗	✗	✗	✗	✗	✗	✗

Ubuntu18.04	AArch64	4.15.0-20-generic	Primary	✓	✓	✓	✓	✓	11.6	✓
	ppc64le	4.15.0-20-generic	Primary	✓	✓	✓	✓	✓	✗	✓
	x86_64	4.15.0-20-generic	Primary	✓	✓	✓	✓	✓	11.6	✓
Ubuntu20.04	AArch64	5.4.0-26-generic	Primary	✓	✓	✓	✓	✓	11.7	✓
	ppc64le	5.4.0-26-generic	Primary	✓	✓	✓	✓	✓	✗	✓
	x86_64	5.4.0-26-generic	Primary	✓	✓	✓	✓	✓	11.7	✓
Ubuntu22.04	AArch64	5.15.0-25-generic	Primary	✓	✗	✓	✓	✓	11.7	✗
	ppc64le	5.15.0-25-generic	Primary	✓	✗	✓	✓	✓	✗	✗
	x86_64	5.15.0-25-generic	Primary	✓	✗	✓	✓	✓	11.7	✗
Ubuntu22.10	x86_64	5.19.0-15-generic	Primary	✓	✗	✓	✓	✗	✗	✗
UOS20.1020	AArch64	4.19.90-2109.1.0.0108.u p2.uel20.aarch64	Primary	✗	✗	✗	✗	✗	✗	✗
	x86_64	4.19.90-2109.1.0.0108.u p2.uel20.x86_64	Primary	✗	✗	✗	✗	✗	✗	✗
UOS20.1040	AArch64	4.19.0-arm64-server	Primary	✗	✗	✗	✗	✗	✗	✗
	x86_64	4.19.0-server-amd64	Primary	✗	✗	✗	✗	✗	✗	✗
Citrix XenServer Host7.1	x86_64	4.4.0+2	Primary	✗	✗	✗	✗	✗	✗	✗
Citrix XenServer Host8.2	x86_64	4.19.0+1	Primary	✗	✗	✗	✗	✗	✗	✗
Kernel 6.0	AArch64	6.0	Primary	✗	✗	✓	✓	✗	✗	✗
	ppc64le	6.0	Primary	✗	✗	✓	✓	✗	✗	✗
	x86_64	6.0	Primary	✗	✗	✓	✓	✗	✗	✗

32 bit platforms are no longer supported in MLNX_EN.

1.4.1 Upgrade/Downgrade Matrix

This section reflects which versions were tested and verified for upgrade and downgrade.

Target Version	Versions Verified for Upgrade/Downgrade	Release Type	Release Date
5.8-5.1.1.2 LTS-Update (June 2024 Update)	5.8-4.1.5.0	LTS-Update	December 2023
	5.8-1.0.1.1	GA-LTS	October 2022
	5.7-1.0.2.0	GA	July 2022
	5.4-3.7.5.0	GA-LTS-Update	November 2022

MLNX_OFED Version Interoperability

This section reflects which versions were tested and verified for multi-version environments.

Target Version	Verified OFED Version Interoperability	Release Type	Release Date
5.8-5.1.1.2 LTS-Update (June Update)	5.8-4.1.5.0	LTS-Update	December 2023

1.4.2 Supported NIC Firmware Versions

As of version 5.1, ConnectX-3, ConnectX-3 Pro or Connect-IB adapter cards are no longer supported. To work with a version that supports these adapter cards, please refer to version 4.9 long-term support (LTS).

This current version is tested with the following NVIDIA adapter card firmware versions:

Adapter Card	Bundled Firmware Version	Additional Supported Firmware Version
BlueField®-2	24.35.4030	24.35.3502
BlueField	18.33.1048	18.33.1048
ConnectX-7	28.35.4030	28.35.3502
ConnectX-6 Lx	26.35.4030	26.35.3502
ConnectX-6 Dx	22.35.4030	22.35.3502
ConnectX-6	20.35.4030	20.35.3502
ConnectX-5/ConnectX-5 Ex	16.35.4030	16.35.3502

ConnectX-4 Lx	14.32.1010	N/A
ConnectX-4	12.28.2006	N/A

For the official firmware versions, please see <https://www.nvidia.com/en-us/networking/> → Support → Support → [Firmware Download](#).

1.5 Changes and New Features

1.5.1 New Features

The following are the new features and changes that were added in this version. The supported adapter cards are specified as follows:

Supported Cards	Description
All HCAs	Supported in the following adapter cards <u>unless specifically stated otherwise:</u> ConnectX-4 / ConnectX -4 Lx / ConnectX-5 / ConnectX-6 / ConnectX-6 Dx / ConnectX-6 Lx / ConnectX-7 / BlueField-2
ConnectX-6 Dx and above	Supported in the following adapter cards <u>unless specifically stated otherwise:</u> ConnectX-6 Dx / ConnectX-6 Lx / ConnectX-7 / BlueField-2
ConnectX-6 and above	Supported in the following adapter cards <u>unless specifically stated otherwise:</u> ConnectX-6 / ConnectX-6 Dx / ConnectX-6 Lx / ConnectX-7 / BlueField-2
ConnectX-5 and above	Supported in the following adapter cards <u>unless specifically stated otherwise:</u> ConnectX-5 / ConnectX-6 / ConnectX-6 Dx / ConnectX-6 Lx / ConnectX-7 / BlueField-2
ConnectX-4 and above	Supported in the following adapter cards <u>unless specifically stated otherwise:</u> ConnectX-4 / ConnectX -4 Lx / ConnectX-5 / ConnectX-6 / ConnectX-6 Dx / ConnectX-6 Lx / ConnectX-7 / BlueField-2

Feature/Change	Description
5.8-5.1.1.2	
Bridge Offload	Bridge Offload in <code>mlx5_core</code> is now enabled by default. To disable it, use the following option in the <code>mlx_ofa_kernel</code> configuration file: <code>"--without-bridge-offload"</code> . To use this option from <code>"mlxofedinstall --add-kernel-support"</code> (kernel modules rebuilding), add the following to the command line: <code>"--kernel-extra-args --without-bridge-offload"</code> .
Bug Fixes	Bug Fixes in This Version

For a list of features from previous version see [Release Notes Change Log History](#) section. For additional information on the new features, please refer to [MLNX_EN User Manual](#).

1.5.2 Unsupported Functionalities/Features/NICs

The following are the unsupported functionalities/features/NICs in the current version:

- Soft-RoCE
- RDMA experimental verbs library (mlnx_lib)
- CIFS (Common Internet File System) module installation.

1.6 Bug Fixes in This Version

Below are the bugs fixed in this version. For a list of fixes previous version, see [Bug Fixes History](#).

Internal Reference Number	Description
3685061	Description: Fixed an issue that could lead to RPM database corruption in RPM-based OSes.
	Keywords: rdmacm
	Discovered in Release: 5.8-4.1.5.0
	Fixed in Release: 5.8-5.1.1.2
3894402	Description: Fixed an overflow issue where RDMA applications could not find the device when creating new RDMA devices.
	Keywords: rdmacm
	Discovered in Release: 5.8-4.1.5.0
	Fixed in Release: 5.8-5.1.1.2
3872233	Description: Fixed an issue that caused a deadlock in the flow of disabling the LAG when changing eswitch mode from switchdev to legacy when a LAG bond existed on the machine.
	Keywords: LAG
	Discovered in Release: 5.8-4.1.5.0
	Fixed in Release: 5.8-5.1.1.2
3845842	Description: Fixed an issue where binding or unbinding PFs may cause memory problems, leading to unexpected kernel behavior.
	Keywords: Physical Functions
	Discovered in Release: 5.8-4.1.5.0
	Fixed in Release: 5.8-5.1.1.2
3817386	Description: Fixed an issue related to the comparison process between the SW steering and FW steering modes to avoid kernel crashes incidences.
	Keywords: SW steering, FW steering
	Discovered in Release: 5.8-4.1.5.0
	Fixed in Release: 5.8-5.1.1.2

1.7 Known Issues

The following is a list of general limitations and known issues of the current version of the release.

Internal Ref. Number	Issue
3711462	Description: The VDPA kernel module may trigger warnings. The logs are harmless and can be discarded by blacklisting this module.
	Keywords: VDPA, warnings
	Workaround: Echo 'blacklist mlx5_vdpa' >/etc/modprobe.d/blacklist_vdpa.conf. Note: this action prevents the inbox module from loading on that system.
	Discovered in Release: 5.8-4.1.5.0
3593433	Description: A Kernel ABI problem in SLES 15SP4 may lead to issues during driver start. This impacts kernels starting from version 5.14.21-150400.24.11.1 up to version 5.14.21-150400.24.63.1 (July 2022 to May 2023), inclusive. For more information, see https://www.suse.com/support/kb/doc/?id=000021137 .
	Keywords: Driver start, SLES
	Workaround: Upgrade to a kernel version newer than 5.14.21-150400.24.63.1 (May 2023).
	Discovered in Release: 5.8-4.1.5.0
3496799	Description: On SLES systems, when installing using the package management method (zypper or "yum") and the package mpi-selector was already installed, zypper will prompt about a change of vendor of the package mpi-selector (a dependency of openmpi).
	Keywords: Installation, SLES
	Workaround: To run an automated install, remove the original mpi-selector package.
	Discovered in Release: 5.8-3.0.7.0
3215514	Description: On EulerOS 2.0SP11, installation with the yum method may fail with an error that mlnx-iproute2 is missing a dependency on libdb-5.3.so()(64bit) .
	Keywords: Installation, EulerOS 2.0SP11, yum
	Workaround: Install in advance the mlnx-iproute2 package with rpm and with the --nodeps option. For example: <code>rpm -Uv --nodeps RPMS/mlnx-iproute2-5.19.0-1.58101.x86_64.rpm</code>
	Discovered in Release: 5.8-1.0.1.1
3247519	Description: On an Ubuntu 22.04 system, when installing using the apt install method to install MLNX_OFED including Open vSwitch, and if the distribution Open vSwitch package was previously installed, the install may fail because of a left-over systemd generated file: the symbolic link /etc/systemd/system/openvswitch-switch.service.requires/ovs-record-hostname.service -> /lib/systemd/system/ovs-record-hostname.service .
	Keywords: Installation, Ubuntu 22.04, Open vSwitch

Internal Ref. Number	Issue
	<p>Workaround: Either remove (and purge: <code>apt purge / dpkg --purge</code>) the package <code>openvswitch-switch</code> before the installation, or, if encountering this error message, remove the file left over and proceed with the installation:</p> <pre>rm file /etc/systemd/system/openvswitch-switch.service.requires/ovs-record-hostname.service apt install -f</pre> <p>Discovered in Release: 5.8-1.0.1.1</p>
3191223	<p>Description: In old kernels, <code>/etc/init.d/openibd stop</code> will fail because of an existing TC rule. Because <code>mlx5_ib</code> is already unloaded, <code>mlx5_core</code> and <code>mlx5_ib</code> will be in an inconsistent state.</p> <p>Keywords: ASAP², eSwitch, TC Rules</p> <p>Workaround: Set eSwitch mode to legacy before enabling SR-IOV or reload <code>mlx5_core</code> to change eSwitch mode to legacy.</p> <p>Discovered in Release: 5.8-1.0.1.1</p>
3199628	<p>Description: <code>ping -6 -i <interface name></code> is broken in v5.18.</p> <p>Keywords: NetDev, -i flag</p> <p>Workaround: In all operating systems that are running Kernel 5.18 and below, remove the -i flag.</p> <p>Discovered in Release: 5.8-1.0.1.1</p>
3002932	<p>Description: Jumbo MTU must be set on all uplinks (i.e., uplinks of <code>*_sf</code> and <code>*_sf_r</code>) at all times.</p> <p>Keywords: NetDev, MTU, Uplink</p> <p>Workaround: Configure jumbo MTU (9216) on all uplink-related interfaces.</p> <p>Discovered in Release: 5.8-1.0.1.1</p>
3228788	<p>Description: Users are advised to avoid running <code>rx-tls-offload</code> over Korg6.0 as its TLS module does not work properly.</p> <p>Keywords: NetDev, TLS</p> <p>Workaround: N/A</p> <p>Discovered in Release: 5.8-1.0.1.1</p>
3130859	<p>Description: The yum install method might be broken on installer regenerated with <code>--add-kernel-support-build-only</code>.</p> <p>Keywords: Installation, yum</p> <p>Workaround: Delete the original <code>mlnx-ofed-all-5.*</code> package and recreate the repository with: <code>createrepo RPMS/</code></p> <p>Discovered in Release: 5.8-1.0.1.1</p>

Internal Ref. Number	Issue
3149387	<p>Description: The package neohost-backend (included in MLNX_OFED) has a strict dependency on Python 2.7 and on the existence of <code>/usr/bin/python</code>. This dependency is because of a pre-installation test (which is a rather non-standard method) for <code>/usr/bin/python</code> will fail the installation if without Python 2.7. As a result, default installation of this on newer systems that do not have a default of Python 2 has been disabled.</p> <p>If there is an explicit request for this installation using the command-line option <code>--with-neohost-backend</code>, this sanity check will be overridden and there will be an attempt to install it regardless. On newer systems, there is likely to not be <code>/usr/bin/python</code> even if Python 2 is installed; as such its installation will fail.</p> <p>Keywords: Installation, Python 2</p> <p>Workaround: If neohost-backend is needed on a newer system, install Python 2 in advance and create the symbolic link <code>/usr/bin/python -> python2</code>.</p> <p>Discovered in Release: 5.8-1.0.1.1</p>
3213777	<p>Description: Oracle Enterprise Linux version 9.0 generates kernel module packages that have dependencies that are not provided by their own kernel RPM packages and thus are not installable.</p> <p>Keywords: Installation, Oracle Enterprise Linux v9.0</p> <p>Workaround: N/A</p> <p>Discovered in Release: 5.8-1.0.1.1</p>
3229904	<p>Description: Restart driver fails to load OFED modules after installing OFED on SLES15sp4 with errata kernel 5.14.21-150400.24.21-default.</p> <p>Keywords: Installation</p> <p>Workaround: Install OFED with <code>--add-kernel-support</code> flag.</p> <p>Discovered in Release: 5.8-1.0.1.1</p>
3189424	<p>Description: VLAN naming is limited to 16 characters (like all other interface names). For names longer than 16 characters, the kernel generates its own interface name VLAN (VID).</p> <p>Keywords: Core, VLAN, Interface Name</p> <p>Workaround: Select a name which complies to the 16-characters limitation.</p> <p>Discovered in Release: 5.8-1.0.1.1</p>
3220855	<p>Description: Creating external SFs on BF ARM when the host (x86) operating system does not support SFs may cause the host to crash.</p> <p>Keywords: Core, Scalable Functions</p> <p>Workaround: N/A</p> <p>Discovered in Release: 5.8-1.0.1.1</p>
3239291	<p>Description: In some topologies, like logical partitions, <code>mlxfwreset</code> is not supported.</p> <p>Keywords: Core, <code>mlxfwreset</code></p> <p>Workaround: N/A</p> <p>Discovered in Release: 5.8-1.0.1.1</p>

Internal Ref. Number	Issue
3114823	<p>Description: The first attempt to create a new iSER connection fails with the following messages in dmesg: iSCSI Login timeout on Network Portal <iSER_Target_IP_ADDR>:3260 isert: isert_get_login_rx: isert_conn 00000000e9239d52 interrupted before got login req</p> <p>After the error, the iSER Initiator connects to the Target successfully, but the memory allocated for the first connection is not freed correctly. As a result, the failed attempt also causes memory leakage.</p> <ul style="list-style-type: none"> • kernel.org Kernel 5.18 • RHEL 9.0 • RHEL 8.6 • Ubuntu 22.04 • SLES 15 SP4 <p>The error happens due to a bug in the scsi_transport_iscsi module, which is not a part of MLNX_EN. As such, the issue cannot be fixed in MLNX_EN. The bug is already fixed in kernel 5.19 by the commit f6eed15f3ea7 ("scsi: iscsi: Exclude zero from the endpoint ID range").</p> <p>Workaround: Update the kernel if the above errors are experienced. If the issue is still reproduced after the kernel update, ask your distro support to apply the bug fix from the upstream kernel.</p> <p>Keywords: iSER Initiator</p> <p>Discovered in Release: 5.7-1.0.2.0</p>
3096911	<p>Description: Installing chkconfig on Rhel9.0 with OFED using yum failed (chkconfig creates /etc/init.d sym link and OFED creates files in this directory, causing a conflict).</p> <p>Workaround: Installing chkconfig before OFED.</p> <p>Keywords: Installation</p> <p>Discovered in Release: 5.7-1.0.2.0</p>
3100544	<p>Description: On a RHEL9.x system, in some cases where inbox modules do not match for the drivers being build, rebuilding the drivers (--add-kernel-support) works, but fails to install the built package, with many errors such as: kernel(__rdma_block_iter_next) = 0x8e7528da is needed by mlnx-ofa_kernel-modules-5.6-OFED.5.6.2.0.9.1.kver.5.14.0_70.13.1.el9_0.aarch64.aarch64 This was caused by a bug in the scripts that creates the Requires and Provides headers that is confused by dependencies between different modules of the same external package.</p> <p>Workaround: dnf install kernel-modules-<kernel-version> # in case it is not the newest.</p> <p>Keywords: Installation, RHEL9.x</p> <p>Discovered in Release: 5.7-1.0.2.0</p>
3132158	<p>Description: Building rdma-core package on Rocky 8.6 OS caused failure in OFED build.</p> <p>Workaround: N/A</p> <p>Keywords: Installation</p> <p>Discovered in Release: 5.7-1.0.2.0</p>
3137440	<p>Description: Python package is missing, need to install it manually.</p> <p>Workaround: Install Python before starting the build.</p> <p>Keywords: Installation, Python</p> <p>Discovered in Release: 5.7-1.0.2.0</p>

Internal Ref. Number	Issue
3141506	<p>Description: kernel-macros package does not support building with KMP enabled. KMP needs to be disabled.</p> <p>Workaround: Build and install MOFED with KMP disabled (without --kmp flag).</p> <p>Keywords: Installation</p> <p>Discovered in Release: 5.7-1.0.2.0</p>
3141506	<p>Description: kernel-macros package does not support building with KMP enabled. KMP needs to be disabled.</p> <p>Workaround: Build and install MOFED with KMP disabled (without --kmp flag).</p> <p>Keywords: Installation</p> <p>Discovered in Release: 5.7-1.0.2.0</p>
3129627	<p>Description: Kernel module packaging is not supported in CtyunOS.</p> <p>Workaround: N/A</p> <p>Keywords: Installation</p> <p>Discovered in Release: 5.7-1.0.2.0</p>
2971708	<p>Description: For OSs in which Devlink supports setting roce-enable/disable, both sysfs roce_enable show and sysfs roce_enable set are disabled, and the RoCE state must be managed exclusively via Devlink. The sysfs interface for roce-enable/disable will be removed entirely for these OSs in a future release. To determine if Devlink can be used to enable or disable RoCE, execute the following console command after starting OFED:</p> <pre data-bbox="424 1131 1390 1193">devlink dev param show grep roce</pre> <p>Devlink supports roce enable/disable if the following line is reflected in the output:</p> <pre data-bbox="424 1279 1390 1341">name enable_roce type generic</pre> <p>For OSs which do not allow enabling/disabling RoCE via Devlink, the sysfs interface behaves as in the previous 2 releases:</p> <ol data-bbox="424 1402 1390 1659" style="list-style-type: none"> 1. For OSs which have Devlink reload, but do not allow setting RoCE state via Devlink: sysfs roce_enable show works, as does sysfs roce_enable set, but Devlink reload must be performed after setting the RoCE state via sysfs in order to activate the desired roce state. 2. For OSs which do not have Devlink reload, RoCE state is managed only by the sysfs interface. 'show' displays the RoCE state and 'set' sets the state and activates it. To determine if Devlink dev reload is supported, execute the following console command (using the bash shell): <pre data-bbox="469 1688 1390 1751">devlink dev help 2>&1 grep reload</pre> <p>Reload is supported if the output is:</p> <pre data-bbox="469 1809 1390 1872">devlink dev reload DEV [netns { PID NAME ID }]</pre> <p>Workaround: N/A</p>

Internal Ref. Number	Issue
	Keywords: Enabling/Disabling RoCE
	Discovered in Release: 5.7-1.0.2.0

Internal Ref. Number	Issue
2971708	<p>Description: For OSs in which Devlink supports setting roce-enable/disable, both sysfs roce_enable show and sysfs roce_enable set are disabled, and the RoCE state must be managed exclusively via Devlink. The sysfs interface for roce-enable/disable will be removed entirely for these OSs in a future release. To determine if Devlink can be used to enable or disable RoCE, execute the following console command after starting OFED:</p> <pre style="border: 1px solid #ccc; padding: 5px;">devlink dev param show grep roce</pre> <p>Devlink supports roce enable/disable if the following line is reflected in the output:</p> <pre style="border: 1px solid #ccc; padding: 5px;">name enable_roce type generic</pre> <p>For OSs which do not allow enabling/disabling RoCE via Devlink, the sysfs interface behaves as in the previous 2 releases:</p> <ol style="list-style-type: none"> For OSs which have Devlink reload, but do not allow setting RoCE state via Devlink: sysfs roce_enable show works, as does sysfs roce_enable set, but Devlink reload must be performed after setting the RoCE state via sysfs in order to activate the desired roce state. For OSs which do not have Devlink reload, RoCE state is managed only by the sysfs interface. 'show' displays the RoCE state and 'set' sets the state and activates it. To determine if Devlink dev reload is supported, execute the following console command (using the bash shell): <pre style="border: 1px solid #ccc; padding: 5px;">devlink dev help 2>&1 grep reload</pre> <p>Reload is supported if the output is:</p> <pre style="border: 1px solid #ccc; padding: 5px;">devlink dev reload DEV [netns { PID NAME ID }]</pre> <p>Workaround: N/A</p> <p>Keywords: Enabling/Disabling RoCE</p> <p>Discovered in Release: 5.7-1.0.2.0</p>
2998194	<p>Description: On some systems with many (e.g., 64) virtual functions (VFs) attached to a ConnectX interface, 'ip link' may give an error message: "Error: Buffer too small for object." This applies to both IP commands: the inbox iproute package in RHEL8.x and the mlnx-iproute2 package from MLNX_OFED. This is known to work well and not give an error in RHEL7.x kernel regardless of what user-space package is used (including user-space from RHEL8.x).</p>

Internal Ref. Number	Issue
	<p>Workaround: N/A</p> <p>Keywords: NetDev, RHEL, Virtual Functions</p> <p>Discovered in Release: 5.6-1.0.3.5</p>
3040350	<p>Description:</p> <ol style="list-style-type: none"> When offload is enabled, removing a physical port from ovs-dpdk bridge requires restarting OVS service. Not doing so will result in wrong configuration of datapath rules. When offload is enabled, the physical port must be attached to a bridge. <p>Workaround:</p> <ol style="list-style-type: none"> When removing a physical port from an ovs-dpdk bridge while offload is enabled, need to restart openvswitch after reattaching it. Attach physical port to a bridge according to the desired topology. <p>Keywords: OVS-DPDK, Bridge, Offload</p> <p>Discovered in Release: 5.6-1.0.3.5</p>
2973726	<p>Description: dec_ttl only work with ConnectX-6. It does not work with ConnectX-5.</p> <p>Workaround: N/A</p> <p>Keywords: OVS-DPDK, dec_ttl</p> <p>Discovered in Release: 5.6-1.0.3.5</p>
2946873	<p>Description: Moving to switchdev mode while deleting namespace may cause a deadlock.</p> <p>Workaround: Unload mlx5_ib module before moving to Switchdev mode.</p> <p>Keywords: ASAP², Switchdev, Namespace</p> <p>Discovered in Release: 5.6-1.0.3.5</p>
2811957	<p>Description: If a system is run from a network boot and is connected to the network storage through an NVIDIA ConnectX card, unloading the mlx5_core driver (such as running '/etc/init.d/openibd restart') will render the system unusable and should therefore be avoided.</p> <p>Workaround: N/A</p> <p>Keywords: Installation, mlx5_core</p> <p>Discovered in Release: 5.6-1.0.3.5</p>
2979243	<p>Description: The kernel in CentOS 7.6alt (for non-x86 architectures) is different than that of RHEL 7.6alt. Some of the MLNX_OFED kernel modules that were built for the RHEL7.6alt kernel will not load on a system with Centos7.6alt kernel. If you want to install MLNX_OFED on such a system, you should use ./mlnxofedinstall --add-kernelsupport to rebuild the kernel modules for the Centos kernel.</p> <p>Workaround: Use add-kernel-support.</p> <p>Keywords: Installation,CentOS</p> <p>Discovered in Release: 5.6-1.0.3.5</p>
3011440	<p>Description: In Debian 11.2, Ubuntu 21.10, and Ubuntu 22.04, attempting to install an "exact" type of metapackage (such as mlnx-ofed-all-exact or mlnx-ofed-basic-exact) may fail with an error regarding the version of mstflint.</p> <p>Workaround: Install also mstflint of the exact same version (e.g., apt install mlnx-ofed-all-exact mstflint=4.16.0-1.56xxxx).</p>

Internal Ref. Number	Issue
	<p>Keywords: Installation, Debian, Ubuntu, MST</p> <p>Discovered in Release: 5.6-1.0.3.5</p>
3024520	<p>Description: The option --copy-ifnames-udev copy some files under /etc (/etc/udev/rules.d/82-net-setup-link.rules and /etc/infiniband/vf-net-link-name.sh) that are never removed--not in the case this option is not given and not upon uninstallation. Those scripts are merely examples. They are files under /etc to be maintained by the user.</p> <p>Workaround: Remove the files, if needed.</p> <p>Keywords: Installation</p> <p>Discovered in Release: 5.6-1.0.3.5</p>
3046601	<p>Description: When rebuilding the kernel modules (--add-kernel-support) for some kernel versions (specifically mainline 4.14) do not unset LDFLAGS properly. Rebuilding xpmem in such a case may fail with the error such as "unrecognized option '-Wl,-z,relro'" in the xpmem build log.</p> <p>Workaround: Either disable building xpmem by adding --without-xpmem to the command line, or edit the kernel Makefile to make it unset LDFLAGS:</p> <pre data-bbox="411 869 1390 931">sed -i -e '/^export ARCH/iLDFLAGS :=' /lib/modules/\$(uname -r)/Makefile</pre> <p>Note: The Makefile may be located elsewhere, such as the top-level directory of the kernel source directory.</p> <p>Keywords: Installation, SLES</p> <p>Discovered in Release: 5.6-1.0.3.5</p>
3046655	<p>Description: A package manager upgrade with zypper (on a SLES system) may prompt a question about vendor change from "Mellanox Technologies" to "OpenFabrics".</p> <p>Workaround: Either accept this when prompted or add the file /etc/zypp/vendors.d/mlnx_ofed with the following content:</p> <pre data-bbox="411 1261 1390 1346">[main] vendors = Mellanox,OpenFabrics</pre> <p>Keywords: Installation, SLES</p> <p>Discovered in Release: 5.6-1.0.3.5</p>
3048411	<p>Description: After installing OFED with rebuilt kernel modules, error messages indicating that the kernel module mlx5_ib failed to load (e.g. "mlx5_ib: Unknown symbol . . .") appear. These messages could be safely ignored because the module eventually loads.</p> <p>Workaround: Run the command 'dracut -f' to update the initramfs.</p> <p>Keywords: Installation</p> <p>Discovered in Release: 5.6-1.0.3.5</p>
3048444	<p>Description: OFED installation failed using yum for --add-kernel-support option (building packages without KMP enabled) if libfabric package is installed.</p> <p>Workaround: Remove libfabric package before OFED installation or use installation script.</p> <p>Keywords: Installation, RHEL 8.5</p> <p>Discovered in Release: 5.6-1.0.3.5</p>

Internal Ref. Number	Issue
3015210	Description: OVS topology where the tunnel device is over a VF and the VF representor is connected to a bond is not supported.
	Workaround: N/A
	Keywords: ASAP ² , ConnectX-6 Dx, Tunnel Over VF, LAG, Connection Tracking
	Discovered in Release: 5.6-1.0.3.5
3028300	Description: OVS metering is not support over kernel 5.17.
	Workaround: N/A
	Keywords: ASAP ² ,OVS, Meter, Kernel 5.17
	Discovered in Release: 5.6-1.0.3.5
3044255	Description: Destroying mlxdevm group while SF is attached to it is not supported.
	Workaround: N/A
	Keywords: ASAP ² , mlxdevm, QoS, Group, Scalable Functions, ConnectX-6 Dx
	Discovered in Release: 5.6-1.0.3.5
3046456	Description: Switching between SwitchDev mode and legacy mode quickly on BlueField-2 can prevent the driver from loading successfully and breaks its health recovery.
	Workaround: Pause 60 seconds between state-altering commands to guarantee the driver health recovery is completed successfully.
	Keywords: ASAP ² , BlueField-2, Health Recovery
	Discovered in Release: 5.6-1.0.3.5
2934149	Description: Adding vDPA ports over ConnectX-5 devices in ovs-dpdk is not supported and will cause a crash.
	Workaround: N/A
	Keywords: OVS-DPDK, ConnectX-5
	Discovered in Release: 5.6-1.0.3.5
2901514	Description: Relaxed Ordering is not working properly on Virtual Functions.
	Workaround: N/A
	Keywords: Relaxed Ordering, VF
	Discovered in Release: 5.6-1.0.3.5

Internal Ref. Number	Issue
2688191	Description: The minimum Tx rate limit is not supported with link speed of 1Gb/s.
	Workaround: N/A
	Keywords: Rate Limit, 1Gb/s
	Discovered in Release: 5.4-1.0.3.0
2870299	Description: Managing SFs is possible using the iproute2 with mlxdevm tool only.
	Workaround: N/A

Internal Ref. Number	Issue
	Keywords: Scalable Functions
	Discovered in Release: 5.5-1.0.3.2
2869722	Description: OFED packages were built with DKMS disabled since building OFED with DKMS failed due to a problem in the DKMS package on UOS. --dkms flag should not be used.
	Workaround: N/A
	Keywords: Installation, DKMS
	Discovered in Release: 5.5-1.0.3.2
2851639	Description: Enabling ARFS in legacy mode and then moving to switchdev mode is not supported and may cause unwanted behavior.
	Workaround: N/A
	Keywords: NetDev, ARFS
	Discovered in Release: 5.5-1.0.3.2
2851639	Description: nvme and iser are not enabled on UOS ARM, because of missing UOS kernel support.
	Workaround: N/A
	Keywords: nvme, iser, UOS ARM
	Discovered in Release: 5.5-1.0.3.2
2860855	Description: Building OFED on RHEL 8.4 with kmp disabled and then installing with yum fails due to some conflicting packages.
	Workaround: Remove libfabric and librpmem packages before OFED installation, or add --allowrasing option to the installation command.
	Keywords: Installation, RHEL 8.4, kmp, yum
	Discovered in Release: 5.5-1.0.3.2
2865983	Description: OFED packages were built with kmp disabled. Building with kmp enabled fails due to missing packages.
	Workaround: N/A
	Keywords: Installation, kmp
	Discovered in Release: 5.5-1.0.3.2

Internal Ref. Number	Issue
2658644	Description: Only match on lower 32 bit of ct_label is supported.
	Workaround: N/A
	Keywords: ASAP ² , Connection Tracking
	Discovered in Release: 5.4-1.0.3.0
2706345	Description: Number of RQ and TIR allocation in the driver depends on total number of MSI-X vectors allocated. Total number of TIRs supported by device is 16K range. Each representor needs number of CPUs worth TIRs, upto maximum of 128.

Internal Ref. Number	Issue
	<p>Workaround: To use large number of VFs, set PF_NUM_PF_MSIX to a smaller value of around 32.</p> <p>Keywords: ASAP², VF, PF_NUM_PF_MSIX</p> <p>Discovered in Release: 5.4-1.0.3.0</p>
2836997	<p>Description: An automatic test that checks a flow meter rate fluctuation stays within a fixed threshold (e.g., 10%) may fail because meter precision is dependent on multiple factors (i.e., rate and burst values and shape of the traffic). To pick the best configuration parameters for a flow meter, perform a couple of test measurements using different values of burst size against expected traffic workload and average the results over an extended period of time (tens of minutes).</p> <p>Workaround: N/A</p> <p>Keywords: ASAP², Meter Threshold</p> <p>Discovered in Release: 5.4-1.0.3.0</p>
2863456	<p>Description: SA limit by packet count (hard and soft) are supported only on traffic originated from the ECPF. Trying to configure them on VF traffic will remove the SA when hard limit is hit, however traffic could still pass as plain text due to the tunnel offload that is used in such configuration.</p> <p>Workaround: N/A</p> <p>Keywords: ASAP², IPsec Full Offload</p> <p>Discovered in Release: 5.4-0.5.1.1</p>
2657392	<p>Description: OFED installation caused CIFS to break in RHEL8.4 and RHEL8.5. A dummy module was added so that CIFS will be disabled after OFED installation in RHEL8.4 and RHEL8.5.</p> <p>Workaround: N/A</p> <p>Keywords: Installation, RHEL8.4, RHEL8.5, CIFS</p> <p>Discovered in Release: 5.4-0.5.1.1</p>
2800993	<p>Description: OpenMPI does not support running across different operating systems and/or CPU architectures.</p> <p>Workaround: N/A</p> <p>Keywords: OpenMPI</p>
2399503	<p>Description: Open vSwitch is not supported on the latest operating systems containing only Python3 support.</p> <p>Workaround: N/A</p> <p>Keywords: Python, Open vSwitch</p>
2657392	<p>Description: OFED installation caused CIFS to break in RHEL8.4. A dummy module was added so that CIFS will be disabled after OFED installation in RHEL8.4.</p> <p>Workaround: N/A</p> <p>Keywords: Installation, RHEL8.4, CIFS</p> <p>Discovered in Release: 5.4-0.5.1.1</p>
2782406	<p>Description: Running yum update will upgrade kylin-release to a higher version. The version of this package is used for kylin10sp2 detection so the script will detect kylin 10 instead of kylin10sp2 and use its repository by mistake.</p>

Internal Ref. Number	Issue
	<p>Workaround: Because there are no special cases for kylin10sp2, the repository that was detected with adding --add-kernel-support to the installation command can be used.</p> <p>Keywords: Upgrade, kylin</p> <p>Discovered in Release: 5.4-3.0.3.0</p>
2755632	<p>Description: On dual port cards with SR-IOV, when one port link is configured to InfiniBand and the other port link is configured to Ethernet, the Ethernet port will not be able to support VST and QinQ.</p> <p>Workaround: N/A</p> <p>Keywords: SR-IOV, VST, QinQ</p> <p>Discovered in Release: 5.4-3.0.3.0</p>
2780436	<p>Description: Non-default MTU (>1500) is not supported with IPsec crypto offload and may cause packet drops.</p> <p>Workaround: N/A</p> <p>Keywords: IPsec, Crypto Offload, MTU</p> <p>Discovered in Release: 5.4-3.0.3.0</p>
2726021	<p>Description: Building packages on openEuler with kmp enabled requires kernel-rpm-macros package installed. kernel-rpm-macros-30-13.oe1 does not support -p option and kernel-rpm-macros-30-18.oe1 should be installed instead. On kylin OS, the version of kernel-rpm-macros package does not support -p option needed to support kmp, so it will stay disabled.</p> <p>Workaround: N/A</p> <p>Keywords: Installation, openEuler</p> <p>Discovered in Release: 5.4-3.0.3.0</p>

Internal Ref. Number	Issue
2750653	<p>Description: Running fragmented traffic in RHEL 8.3 (4.18.0-240.el8.x86_64) may cause call trace in build_skb.</p> <p>Workaround: Update to RHEL 8.3 z-stream 4.18.0-240.22.1.el8_3.x86_64.</p> <p>Keywords: RHEL 8.3, Kernel Panic, Call Trace, fr</p> <p>Discovered in Release: 5.4-1.0.3.0</p>
2629375	<p>Description: Matching on CT label is only supported when matching on lower 32 bits. Full match on all 128 bits of CT label is not supported.</p> <p>Workaround: N/A</p> <p>Keywords: ASAP², Connection Tracking, Label</p> <p>Discovered in Release: 5.4-1.0.3.0</p>
2707997	<p>Description: Installation in the package manager mode under SLES 15.x may require user-intervention if the original libibverbs is installed.</p>

Internal Ref. Number	Issue
	<p>Workaround: zypper install --force-resolution mlnx-ofed-all</p> <p>Keywords: Installation, libibverbs</p> <p>Discovered in Release: 5.4-1.0.3.0</p>
2708531	<p>Description: Installation in the package manager mode under SLES 15.x may require user-intervention if the original libopenvswitch is installed.</p> <p>Workaround: zypper install --force-resolution mlnx-ofed-all</p> <p>Keywords: Installation</p> <p>Discovered in Release: 5.4-1.0.3.0</p>
2703043	<p>Description: Congested TCP lock for kTLS TX device offload traffic compromises the performance.</p> <p>Workaround: Disable TCP selective acknowledgement: echo 0 > /proc/sys/net/ipv4/tcp_sack</p> <p>Keywords: kTLS TX</p> <p>Discovered in Release: 5.4-1.0.3.0</p>
2676405	<p>Description: If the package interface-rename is active (on XenServer, for example), the interface renaming by the OFED will not be done to eliminate conflicts.</p> <p>Workaround: N/A</p> <p>Keywords: Interface Renaming</p> <p>Discovered in Release: 5.4-1.0.3.0</p>
2687943	<p>Description: Offload of rules which redirect from VF on one PF to VF on second PF is not supported on socket-direct devices.</p> <p>Workaround: N/A</p> <p>Keywords: ASAP², Socket-Direct</p> <p>Discovered in Release: 5.4-1.0.3.0</p>
2678672	<p>Description: When disabling switchdev mode, the qdisc in tunnel device cannot be destroyed and mlx5e_stats_flower() is still called by OVS resulting in NULL pointer panic and memory leak.</p> <p>Workaround: N/A</p> <p>Keywords: SwitchDev, mlx5, Tunnel Traffic</p> <p>Discovered in Release: 5.4-1.0.3.0</p>
2566548	<p>Description: On PPC systems when EEH is enabled, running fw sync reset (either by mlx5fwreset with flag --sync 1 or by devlink dev reload action fw_activate), the EEH may catch the PCI reset and take ownership on the flow. When run few times in sequence, the EEH may also decide to disable the device.</p> <p>Workaround: Administrator may disable EEH before running firmware sync reset on the device.</p> <p>Keywords: PPC, EEH</p> <p>Discovered in Release: 5.4-1.0.3.0</p>
2617950	<p>Description: TX port timestamp feature is supported for kernel versions 3.15 and greater. On older kernel versions, the feature will not be supported and ptp_tx<X>_* counters will not increment.</p>

Internal Ref. Number	Issue
	Workaround: N/A
	Keywords: Ethtool
	Discovered in Release: 5.4-1.0.3.0
2390731	Description: Ethtool does not display Port Speed advertised/capability above 100Gb/s over and below kernels 5.0, even when supported.
	Workaround: N/A
	Keywords: Ethtool, Port Speed
	Discovered in Release: 5.4-1.0.3.0

Internal Ref. Number	Issue
2585575	Description: After disabling sync reset by setting enable_remote_dev_reset to false, running firmware sync reset a few times may lead to general protection fault and system may get stuck.
	Workaround: N/A
	Keywords: Firmware Upgrade
	Discovered in Release: 5.3-1.0.0.1
2582565	Description: Conducting a firmware reset or unbinding the PF while in switchdev mode may cause a kernel crash.
	Workaround: N/A
	Keywords: SwitchDev, ASAP ² , Unbind, Firmware Reset
	Discovered in Release: 5.3-1.0.0.1
2587802	Description: PTP synchronization may be lost while using tx_port_ts private flag.
	Workaround: Toggle private flag: ethtool --set-priv-flags <ifs> tx_port_ts off ethtool --set-priv-flags <ifs> tx_port_ts on restart ptp4l application
	Keywords: PTP Synchronization
	Discovered in Release: 5.3-1.0.0.1
2574943	Description: When running kernel 5.8 and bellow or RHEL 8.2 and below, sampled packets do not support tunnel information.
	Workaround: N/A
	Keywords: ASAP ² , sFLOW
	Discovered in Release: 5.3-1.0.0.1
2568417	Description: Upon upgrade to version 5.3, the package manager tool will install the new packages and then remove the old packages, a depmod WARNING on "mlx5_fpga_tools" will appear. This warning can be safely ignored. mlx5_fpga_tools is a module that existed in version 5.2 and was removed in 5.3.
	Workaround: N/A
	Keywords: Upgrade; mlx5_fpga_tools

Internal Ref. Number	Issue
	Discovered in Release: 5.3-1.0.0.1
2506425	<p>Description: When installing kmod packages on EulerOS 2.0SP9 or OpenEuler 20.03, the following error appears: "modprobe: FATAL: could not get modversions of <directory>". This error can be safely ignored. It is caused by incorrectly adding directories to a list of modules processed by /usr/sbin/weak-modules.</p> <p>Workaround: N/A</p> <p>Keywords: Installation; modules; kmod</p> <p>Discovered in Release: 5.3-1.0.0.1</p>
2492509	<p>Description: When installing the driver on OpenEuler or on EulerOS 2.0SP9, rebuilding the drivers (--add-kernel-support) with the --kmp option (to create kmod packages) generates packages that are uninstallable because they have a dependency on "/sbin/depmod" that the system does not provide. This dependency is created by a buggy kmod package building tool included with the distribution.</p> <p>Workaround: N/A</p> <p>Keywords: add-kernel-support</p> <p>Discovered in Release: 5.3-1.0.0.1</p>
2479327	<p>Description: On SLES 12 SP5, if the kernel was upgraded to 4.12.14-122.46, it is not possible to rebuild kernel modules (--add-kernel-support) without upgrading gcc as well to at least 4.8.5-31.23.2.</p> <p>Workaround: N/A</p> <p>Keywords: Upgrade; SLES 12; add-kernel-support</p> <p>Discovered in Release: 5.3-1.0.0.1</p>
2584441	<p>Description: On SLES 12 SP5, if the kernel was upgraded to 4.12.14-122.46, it is not possible to rebuild kernel modules (--add-kernel-support) without upgrading gcc as well to at least 4.8.5-31.23.2.</p> <p>Workaround: N/A</p> <p>Keywords: Upgrade; SLES 12; add-kernel-support</p> <p>Discovered in Release: 5.3-1.0.0.1</p>
2460865	<p>Description: When setting MTU to low values, such as 68 bytes, packets may fail on oversize.</p> <p>Workaround: N/A</p> <p>Keywords: MTU</p> <p>Discovered in Release: 5.3-1.0.0.1</p>
2383318	<p>Description: On kernels based on RedHat 7.2, the "tx_port_ts" feature, as set by ethtool --set-priv-flags, is disabled.</p> <p>Workaround: N/A</p> <p>Keywords: RedHat; tx_port_ts</p> <p>Discovered in Release: 5.3-1.0.0.1</p>
2575647	<p>Description: An OvS-DPDK crash might occur while doing live-migration for VMs that use virtio-interfaces that are accelerated using OvS-DPDK vDPA ports.</p> <p>Workaround: N/A</p>

Internal Ref. Number	Issue
	Keywords: OvS-DPDK vDPA, Live-migration
	Discovered in Release: 5.3-1.0.0.1

Internal Ref. Number	Issue
2395082	Description: A call trace may take place when moving from SwitchDev mode back to Legacy mode in Kernel v5.9 due to a kernel issue in tcf_block_unbind.
	Workaround: N/A
	Keywords: ASAP ² ;SwitchDev; call trace; kernel; tcf_block_unbind
	Discovered in Release: 5.2-1.0.4.0

Internal Ref. Number	Issue
2209987	Description: aRFS feature (activated using "ethtool ntuple on") is disabled for kernel 4.1 or below.
	Workaround: N/A
	Keywords: aRFS
	Discovered in Release: 5.1-1.0.4.0
2248996	Description: Downgrading the firmware version for ConnectX-6 cards using " <code>install --fw-update-only --force-fw-update</code> " fails.
	Workaround: Manually downgrade the firmware version - please see Firmware Update Instructions .
	Keywords: Firmware, ConnectX-6
	Discovered in Release: 5.1-1.0.4.0
2175930	Description: When using MLNX_EN v5.1 on PPC architectures with kernels v5.5 or v5.6 and an old ethtool utility, a harmless warning call trace may appear in the dmesg due to mismatch between user space and kernel. The warning call trace mentions ethtool_notify.
	Workaround: Update the ethtool utility to version 5.6 on such systems in order to avoid the call trace.
	Keywords: PPC, ethtool_notify, kernel
	Discovered in Release: 5.1-1.0.4.0
2198764	Description: If MLNX_EN is installed on a Debian or Ubuntu system that is run in chroot environment, the openibd service will not be enabled. If the chroot files are being used as a base of a full system, the openibd service is left disabled.
	Workaround: Currently, openibd is a sysv-init script that you can enable manually by running: <code>update-rc.d openibd defaults</code>
	Keywords: chroot, Debian , Ubuntu, openibd
	Discovered in Release: 5.1-1.0.4.0

Internal Ref. Number	Issue
2237134	<p>Description: Running connection tracking (CT) with FW steering may cause CREATE_FLOW_TABLE command to fail with syndrome.</p> <p>Workaround: Configure OVS to use a single handler-thread: <pre>#ovs-vsctl set Open_vSwitch . other_config:n-handler-threads=1</pre></p> <p>Keywords: Connection tracking, ASAP, OVS, FW steering</p> <p>Discovered in Release: 5.1-1.0.4.0</p>
2239894	<p>Description: Running OpenVSwitch offload with high traffic throughput can cause low insertion rate due to high CPU usage.</p> <p>Workaround: Reduce the number of combined channels of the uplink using "ethtool -L".</p> <p>Keywords: Insertion rate, ASAP2</p> <p>Discovered in Release: 5.1-1.0.4.0</p>
2240671	<p>Description: Header rewrite action is not supported over RHEL/CentOS 7.4.</p> <p>Workaround: N/A</p> <p>Keywords: ASAP, header rewrite, RHEL, RedHat, CentOS, OS</p> <p>Discovered in Release: 5.1-1.0.4.0</p>
2242546	<p>Description: Tunnel offload (encap/decap) may cause kernel panic if nf_tables module is not probed.</p> <p>Workaround: Make sure to probe the nf_tables module before inserting any rule.</p> <p>Keywords: Kernel v5.7, ASAP, kernel panic</p> <p>Discovered in Release: 5.1-1.0.4.0</p>
2143007	<p>Description: IPsec packets are dropped during heavy traffic due to a bug in net/xfrm Linux Kernel.</p> <p>Workaround: Make sure the Kernel is modified to apply the following patch: "xfrm: Fix double ESP trailer insertion in IPsec crypto offload".</p> <p>Keywords: IPsec, xfrm</p> <p>Discovered in Release: 5.1-1.0.4.0</p>
2225952	<p>Description: VF mirroring with TC policy skip_sw is not supported on RHEL/CentOS 7.4, 7.5 and 7.6 OSs.</p> <p>Workaround: N/A</p> <p>Keywords: ASAP², Mirroring, RHEL, RedHat, OS</p> <p>Discovered in Release: 5.1-1.0.4.0</p>
2216521	<p>Description: After upgrading MLNX_EN from v5.0 or earlier, ibdev2netdev utility changes the installation prefix to /usr/sbin. Therefore, it cannot be found while found in the same SHELL environment.</p> <p>Workaround: After installing MLNX_EN, log out and log in again to refresh the SHELL environment.</p> <p>Keywords: ibdev2netdev</p> <p>Discovered in Release: 5.1-1.0.4.0</p>

Internal Ref. Number	Issue
2202520	Description: Rules with VLAN push/pop, encap/decap and header rewrite actions together are not supported.
	Workaround: N/A
	Keywords: ASAP ² , SwitchDev, VLAN push/pop, encap/decap, header rewrite
	Discovered in Release: 5.1-1.0.4.0
2210752	Description: Switching from Legacy mode to SwitchDev mode and vice-versa while TC rules exist on the NIC will result in failure.
	Workaround: Before attempting to switch mode, make sure to delete all TC rules on the NIC or stop OpenvSwitch.
	Keywords: ASAP ² , Devlink, Legacy SR-IOV
	Discovered in Release: 5.1-1.0.4.0
2125036/2125031	Description: Upgrading the MLNX_EN from an UPSTREAM_LIBS based version to an MLNX_LIBS based version fails unless the driver is uninstalled and then re-installed.
	Workaround: Make sure to uninstall and re-install MLNX_EN to complete the upgrade.
	Keywords: Installation, UPSTREAM_LIBS, MLNX_LIBS
	Discovered in Release: 5.1-1.0.4.0
2105447	Description: hns_roce warning messages will appear in the dmesg after reboot on Euler2 SP3 OSs.
	Workaround: N/A
	Keywords: hns_roce, dmesg, Euler
	Discovered in Release: 5.1-1.0.4.0
2112251	Description: On kernels 4.10-4.14, when Geneve tunnel's remote endpoint is defined using IPv6, packets larger than MTU are not fragmented, resulting in no traffic sent.
	Workaround: Define geneve tunnel's remote endpoint using IPv4.
	Keywords: Kernel, Geneve, IPv4, IPv6, MTU, fragmentation
	Discovered in Release: 5.1-1.0.4.0
2102902	Description: A kernel panic may occur over RH8.0-4.18.0-80.el8.x86_64 OS when opening kTLS offload connection due to a bug in kernel TLS stack.
	Workaround: N/A
	Keywords: TLS offload, mlx5e
	Discovered in Release: 5.1-1.0.4.0
2111534	Description: A Kernel panic may occur over Ubuntu19.04-5.0.0-38-generic OS when opening kTLS offload connection due to a bug in the Kernel TLS stack.
	Workaround: N/A
	Keywords: TLS offload, mlx5e
	Discovered in Release: 5.1-1.0.4.0

Internal Ref. Number	Issue
2094176	<p>Description: When running in a large scale in VF-LAG mode, bandwidth may be unstable.</p> <p>Workaround: N/A</p> <p>Keywords: VF LAG</p> <p>Discovered in Release: 5.0-1.0.0.0</p>
2044544	<p>Description: When working with OSs with Kernel v4.10, bonding module does not allow setting MTUs larger than 1500 on a bonding interface.</p> <p>Workaround: Upgrade your Kernel version to v4.11 or above.</p> <p>Keywords: Bonding, MTU, Kernel</p> <p>Discovered in Release: 5.0-1.0.0.0</p>
1882932	<p>Description: Libibverbs dependencies are removed during OFED installation, requiring manual installation of libraries that OFED does not reinstall.</p> <p>Workaround: Manually install missing packages.</p> <p>Keywords: libibverbs, installation</p> <p>Discovered in Release: 5.0-1.0.0.0</p>
2058535	<p>Description: ibdev2netdev command returns duplicate devices with different ports in SwitchDev mode.</p> <p>Workaround: Use /opt/mellanox/iproute2/sbin/rdma link show command instead.</p> <p>Keywords: ibdev2netdev</p> <p>Discovered in Release: 5.0-1.0.0.0</p>
2072568	<p>Description: In RHEL/CentOS 7.2 OSs, adding drop rules when <code>act_gact</code> is not loaded may cause a kernel crash.</p> <p>Workaround: Preload all needed modules to avoid such a scenario (<code>cls_flower</code>, <code>act_mirred</code>, <code>act_gact</code>, <code>act_tunnel_key</code> and <code>act_vlan</code>).</p> <p>Keywords: RHEL/CentOS 7.2, Kernel 4.9, call trace, ASAP</p> <p>Discovered in Release: 5.0-1.0.0.0</p>
2093698	<p>Description: VF LAG configuration is not supported when the NUM_OF_VFS configured in mlxconfig is higher than 64.</p> <p>Workaround: N/A</p> <p>Keywords: VF LAG, SwitchDev mode, ASAP</p> <p>Discovered in Release: 5.0-1.0.0.0</p>
2093746	<p>Description: Devlink health dumps are not supported on kernels lower than v5.3.</p> <p>Workaround: N/A</p> <p>Keywords: Devlink, health report, dump</p> <p>Discovered in Release: 5.0-1.0.0.0</p>
2083427	<p>Description: For kernels with connection tracking support, neigh update events are not supported, requiring users to have static ARPs to work with OVS and VxLAN.</p> <p>Workaround: N/A</p> <p>Keywords: VxLAN, VF LAG, neigh, ARP</p>

Internal Ref. Number	Issue
	Discovered in Release: 5.0-1.0.0.0
2067012	Description: MLNX_EN cannot be installed on Debian 9.11 OS in SwitchDev mode.
	Workaround: Install OFED with the flag <code>--add-kernel-support</code> .
	Keywords: ASAP, SwitchDev, Debian, Kernel
	Discovered in Release: 5.0-1.0.0.0
2036572	Description: When using a thread domain and the lockless rdma-core ibv_post_send path, there is an additional CPU penalty due to required barriers around the device MMIO buffer that were omitted in MLNX_EN.
	Workaround: N/A
	Keywords: rdma-core, write-combining, MMIO buffer
	Discovered in Release: 5.0-1.0.0.0

Internal Ref. Number	Issue
-	Description: The argparse module is installed by default in Python versions $\Rightarrow 2.7$ and ≥ 3.2 . In case an older Python version is used, the argparse module is not installed by default.
	Workaround: Install the argparse module manually.
	Keywords: Python, MFT, argparse, installation
	Discovered in Release: 4.7-3.2.9.0
1997230	Description: Running mlxfwreset or unloading mlx5_core module while contrak flows are offloaded may cause a call trace in the kernel.
	Workaround: Stop OVS service before calling mlxfwreset or unloading mlx5_core module.
	Keywords: Contrak, ASAP, OVS, mlxfwrest, unload
	Discovered in Release: 4.7-3.2.9.0
1955352	Description: Moving 2 ports to SwitchDev mode in parallel is not supported.
	Workaround: N/A
	Keywords: ASAP, SwitchDev
	Discovered in Release: 4.7-3.2.9.0
1979958	Description: VxLAN IPv6 offload is not supported over CentOS/RHEL v7.2 OSs.
	Workaround: N/A
	Keywords: Tunnel, VXLAN, ASAP, IPv6
	Discovered in Release: 4.7-3.2.9.0
1991710	Description: PRIO_TAG_REQUIRED_EN configuration is not supported and may cause call trace.
	Workaround: N/A
	Keywords: ASAP, PRIO_TAG, mstconfig
	Discovered in Release: 4.7-3.2.9.0

Internal Ref. Number	Issue
1967866	Description: Enabling ECMP offload requires the VFs to be unbound and VMs to be shut down.
	Workaround: N/A
	Keywords: ECMP, Multipath, ASAP ²
	Discovered in Release: 4.7-3.2.9.0
1821235	Description: When using mlx5dv_dr API for flow creation, for flows which execute the "encapsulation" action or "push vlan" action, metadata C registers will be reset to zero.
	Workaround: Use the both actions at the end of the flow process.
	Keywords: Flow steering
	Discovered in Release: 4.7-1.0.0.1
1921981	Description: On Ubuntu, Debian and RedHat 8 and above OSS, parsing the mfa2 file using the mstarchive might result in a segmentation fault.
	Workaround: Use mlxarchive to parse the mfa2 file instead.
	Keywords: MFT, mfa2, mstarchive, mlxarchive, Ubuntu, Debian, RedHat, operating system
	Discovered in Release: 4.7-1.0.0.1
1840288	Description: MLNX_EN does not support XDP features on RedHat 7 OS, despite the declared support by RedHat.
	Workaround: N/A
	Keywords: XDP, RedHat
	Discovered in Release: 4.7-1.0.0.1
1892663	Description: mlnx_tune script does not support python3 interpreter.
	Workaround: Run mlnx_tune with python2 interpreter only.
	Keywords: mlnx_tune, python3, python2
	Discovered in Release: 4.7-1.0.0.1

Internal Ref. Number	Issue
1753629	Description: A bonding bug found in Kernels 4.12 and 4.13 may cause a slave to become permanently stuck in <code>BOND_LINK_FAIL</code> state. As a result, the following message may appear in dmesg: <code>bond: link status down for interface eth1, disabling it in 100 ms</code>
	Workaround: N/A
	Keywords: Bonding, slave
	Discovered in Release: 4.6-1.0.1.1
1712068	Description: Uninstalling MLNX_EN automatically results in the uninstallation of several libraries that are included in the MLNX_EN package, such as InfiniBand-related libraries.
	Workaround: If these libraries are required, reinstall them using the local package manager (yum/dnf).
	Keywords: MLNX_EN libraries

Internal Ref. Number	Issue
	Discovered in Release: 4.6-1.0.1.1
-	<p>Description: Due to changes in libraries, MFT v4.11.0 and below are not forward compatible with MLNX_EN v4.6-1.0.0.0 and above. Therefore, with MLNX_EN v4.6-1.0.0.0 and above, it is recommended to use MFT v4.12.0 and above.</p> <p>Workaround: N/A</p> <p>Keywords: MFT compatible</p> <p>Discovered in Release: 4.6-1.0.1.1</p>
1730840	<p>Description: On ConnectX-4 HCAs, GID index for RoCE v2 is inconsistent when toggling between enabled and disabled interface modes.</p> <p>Workaround: N/A</p> <p>Keywords: RoCE v2, GID</p> <p>Discovered in Release: 4.6-1.0.1.1</p>
1717428	<p>Description: On kernels 4.10-4.14, MTUs larger than 1500 cannot be set for a GRE interface with any driver (IPv4 or IPv6).</p> <p>Workaround: Upgrade your kernel to any version higher than v4.14.</p> <p>Keywords: Fedora 27, gretap, ip_gre, ip_tunnel, ip6_gre, ip6_tunnel</p> <p>Discovered in Release: 4.6-1.0.1.1</p>
1748343	<p>Description: Driver reload takes several minutes when a large number of VFs exists.</p> <p>Workaround: N/A</p> <p>Keywords: VF, SR-IOV</p> <p>Discovered in Release: 4.6-1.0.1.1</p>
1733974	<p>Description: Running heavy traffic (such as 'ping flood') while bringing up and down other mlx5 interfaces may result in "INFO: rcu_preempt detected stalls on CPUs/tasks:" call traces.</p> <p>Workaround: N/A</p> <p>Keywords: mlx5</p> <p>Discovered in Release: 4.6-1.0.1.1</p>
-	<p>Description: On ConnectX-6 HCAs and above, an attempt to configure advertisement (any bitmap) will result in advertising the whole capabilities.</p> <p>Workaround: N/A</p> <p>Keywords: 200GbE, advertisement, Ethtool</p> <p>Discovered in Release: 4.6-1.0.1.1</p>

Internal Ref. Number	Issue
581631	<p>Description: GID entries referenced to by a certain user application cannot be deleted while that user application is running.</p> <p>Workaround: N/A</p>

Internal Ref. Number	Issue
	Keywords: RoCE, GID
	Discovered in Release: 4.5-1.0.1.0
1403313	Description: Attempting to allocate an excessive number of VFs per PF in operating systems with kernel versions below v4.15 might fail due to a known issue in the Kernel.
	Workaround: Make sure to update the Kernel version to v4.15 or above.
	Keywords: VF, PF, IOMMU, Kernel, OS
	Discovered in Release: 4.5-1.0.1.0
1521877	Description: On SLES 12 SP1 OSs, a kernel tracepoint issue may cause undefined behavior when inserting a kernel module with a wrong parameter.
	Workaround: N/A
	Keywords: mlx5 driver, SLES 12 SP1
	Discovered in Release: 4.5-1.0.1.0

Internal Ref. Number	Issue
504073	Description: When using ConnectX-5 with LRO over PPC systems, the HCA might experience back pressure due to delayed PCI Write operations. In this case, bandwidth might drop from line-rate to ~35Gb/s. Packet loss or pause frames might also be observed.
	Workaround: Look for an indication of PCI back pressure (“outbound_pci_stalled_wr” counter in ethtools advancing). Disabling LRO helps reduce the back pressure and its effects.
	Keywords: Flow Control, LRO
	Discovered in Release: 4.4-1.0.0.0
1424233	Description: On RHEL v7.3, 7.4 and 7.5 OSs, setting IPv4-IP-forwarding will turn off LRO on existing interfaces. Turning LRO back on manually using ethtool and adding a VLAN interface may cause a warning call trace.
	Workaround: Make sure IPv4-IP-forwarding and LRO are not turned on at the same time.
	Keywords: IPv4 forwarding, LRO
	Discovered in Release: 4.4-1.0.1.0
1442507	Description: Retpoline support in GCC causes an increase in CPU utilization, which results in IP forwarding’s 15% performance drop.
	Workaround: N/A
	Keywords: Retpoline, GCC, CPU, IP forwarding, Spectre attack
	Discovered in Release: 4.4-1.0.1.0
1425129	Description: MLNX_EN cannot be installed on SLES 15 OSs using Zypper repository.
	Workaround: Install MLNX_EN using the standard installation script instead of Zypper repository.
	Keywords: Installation, SLES, Zypper
	Discovered in Release: 4.4-1.0.1.0

Internal Ref. Number	Issue
1241056	<p>Description: When working with ConnectX-4/ConnectX-5 HCAs on PPC systems with Hardware LRO and Adaptive Rx support, bandwidth drops from full wire speed (FWS) to ~60Gb/s.</p> <p>Workaround: Make sure to disable Adaptive Rx when enabling Hardware LRO: <i>ethtool -C <interface> adaptive-rx off</i> <i>ethtool -C <interface> rx-usecs 8 rx-frames 128</i></p> <p>Keywords: Hardware LRO, Adaptive Rx, PPC</p> <p>Discovered in Release: 4.3-1.0.1.0</p>
1090612	<p>Description: NVMeoF protocol does not support LBA format with non-zero metadata size. Therefore, NVMe namespace configured to LBA format with metadata size bigger than 0 will cause Enhanced Error Handling (EEH) in PowerPC systems.</p> <p>Workaround: Configure the NVMe namespace to use LBA format with zero sized metadata.</p> <p>Keywords: NVMeoF, PowerPC, EEH</p> <p>Discovered in Release: 4.3-1.0.1.0</p>
1309621	<p>Description: In switchdev mode default configuration, stateless offloads/steering based on inner headers is not supported.</p> <p>Workaround: To enable stateless offloads/steering based on inner headers, disable encap by running: <i>devlink dev eswitch show pci/0000:83:00.1 encap disable</i> Or, in case devlink is not supported by the kernel, run: <i>echo none > /sys/kernel/debug/mlx5/<BDF>/compat/encap</i></p> <p>Note: This is a hardware-related limitation.</p> <p>Keywords: switchdev, stateless offload, steering</p> <p>Discovered in Release: 4.3-1.0.1.0</p>
1275082	<p>Description: When setting a non-default IPv6 link local address or an address that is not based on the device MAC, connection establishments over RoCEv2 might fail.</p> <p>Workaround: N/A</p> <p>Keywords: IPV6, RoCE, link local address</p> <p>Discovered in Release: 4.3-1.0.1.0</p>
1307336	<p>Description: In RoCE LAG mode, when running <i>ibdev2netdev -v</i>, the port state of the second port of the mlx4_0 IB device will read "NA" since this IB device does not have a second port.</p> <p>Workaround: N/A</p> <p>Keywords: mlx4, RoCE LAG, ibdev2netdev, bonding</p> <p>Discovered in Release: 4.3-1.0.1.0</p>
1296355	<p>Description: Number of MSI-X that can be allocated for VFs and PFs in total is limited to 2300 on Power9 platforms.</p> <p>Workaround: N/A</p> <p>Keywords: MSI-X, VF, PF, PPC, SR-IOV</p> <p>Discovered in Release: 4.3-1.0.1.0</p>

Internal Ref. Number	Issue
1259293	<p>Description: On Fedora 20 operating systems, driver load fails with an error message such as: “ [185.262460] kmem_cache_sanity_check (fs_ftes_0000:00:06.0): Cache name already exists. ” This is caused by SLUB allocators grouping multiple slab kmem_cache_create into one slab cache alias to save memory and increase cache hotness. This results in the slab name to be considered stale.</p> <p>Workaround: Upgrade the kernel version to kernel-3.19.8-100.fc20.x86_64. Note that after rebooting to the new kernel, you will need to rebuild MLNX_EN against the new kernel version.</p> <p>Keywords: Fedora, driver load</p> <p>Discovered in Release: 4.3-1.0.1.0</p>
1264359	<p>Description: When running perf test (ib_send_bw, ib_write_bw, etc.) in rdma-cm mode, the resp_cqe_error counter under /sys/class/infiniband/mlx5_0/ports/1/hw_counters/resp_cqe_error might increase. This behavior is expected and it is a result of receive WQEs that were not consumed.</p> <p>Workaround: N/A</p> <p>Keywords: perf test, RDMA CM, mlx5</p> <p>Discovered in Release: 4.3-1.0.1.0</p>
1264956	<p>Description: Configuring SR-IOV after disabling RoCE LAG using sysfs (/sys/bus/pci/drivers/mlx5_core/<bdf>/roce_lag_enable) might result in RoCE LAG being enabled again in case SR-IOV configuration fails.</p> <p>Workaround: Make sure to disable RoCE LAG once again.</p> <p>Keywords: RoCE LAG, SR-IOV</p> <p>Discovered in Release: 4.3-1.0.1.0</p>

Internal Ref. Number	Issue
1263043	<p>Description: On RHEL7.4, due to an OS issue introduced in kmod package version 20-15.el7_4.6, parsing the depmod configuration files will fail, resulting in either of the following issues:</p> <ul style="list-style-type: none"> • Driver restart failure prompting an error message, such as: “ <i>ERROR: Module mlx5_core belong to kernel which is not a part of MLNX_EN, skipping..</i> ” • nvmet_rdma kernel module dysfunction, despite installing MLNX_EN using the "--with-nvmf" option. An error message, such as: “ <i>nvmet_rdma: unknown parameter 'offload_mem_start' ignored</i> ” will be seen in dmesg output <p>Workaround: Go to RedHat webpage to upgrade the kmod package version.</p> <p>Keywords: driver restart, kmod, kmp, nvmf, nvmet_rdma</p> <p>Discovered in Release: 4.2-1.2.0.0</p>

Internal Ref. Number	Issue
-	<p>Description: Packet Size (Actual Packet MTU) limitation for IPsec offload on Innova IPsec adapter cards: The current offload implementation does not support IP fragmentation. The original packet size should be such that it does not exceed the interface's MTU size after the ESP transformation (encryption of the original IP packet which increases its length) and the headers (outer IP header) are added:</p> <ul style="list-style-type: none"> • Inner IP packet size \leq I/F MTU - ESP additions (20) - outer_IP (20) - fragmentation issue reserved length (56) • Inner IP packet size \leq I/F MTU - 96 <p>This mostly affects forwarded traffic into smaller MTU, as well as UDP traffic. TCP does PMTU discovery by default and clamps the MSS accordingly.</p> <p>Workaround: N/A</p> <p>Keywords: Innova IPsec, MTU</p> <p>Discovered in Release: 4.2-1.0.1.0</p>
-	<p>Description: No LLC/SNAP support on Innova IPsec adapter cards.</p> <p>Workaround: N/A</p> <p>Keywords: Innova IPsec, LLC/SNAP</p> <p>Discovered in Release: 4.2-1.0.1.0</p>
-	<p>Description: No support for FEC on Innova IPsec adapter cards. When using switches, there may be a need to change its configuration.</p> <p>Workaround: N/A</p> <p>Keywords: Innova IPsec, FEC</p> <p>Discovered in Release: 4.2-1.0.1.0</p>
955929	<p>Description: Heavy traffic may cause SYN flooding when using Innova IPsec adapter cards.</p> <p>Workaround: N/A</p> <p>Keywords: Innova IPsec, SYN flooding</p> <p>Discovered in Release: 4.2-1.0.1.0</p>
-	<p>Description: Priority Based Flow Control is not supported on Innova IPsec adapter cards.</p> <p>Workaround: N/A</p> <p>Keywords: Innova IPsec, Priority Based Flow Control</p> <p>Discovered in Release: 4.2-1.0.1.0</p>
-	<p>Description: Pause configuration is not supported when using Innova IPsec adapter cards. Default pause is global pause (enabled).</p> <p>Workaround: N/A</p> <p>Keywords: Innova IPsec, Global pause</p> <p>Discovered in Release: 4.2-1.0.1.0</p>
1045097	<p>Description: Connecting and disconnecting a cable several times may cause a link up failure when using Innova IPsec adapter cards.</p> <p>Workaround: N/A</p> <p>Keywords: Innova IPsec, Cable, link up</p> <p>Discovered in Release: 4.2-1.0.1.0</p>

Internal Ref. Number	Issue
-	<p>Description: On Innova IPsec adapter cards, supported MTU is between 512 and 2012 bytes. Setting MTU values outside this range might fail or might cause traffic loss.</p> <p>Workaround: Set MTU between 512 and 2012 bytes.</p> <p>Keywords: Innova IPsec, MTU</p> <p>Discovered in Release: 4.2-1.0.1.0</p>
1125184	<p>Description: In old kernel versions, such as Ubuntu 14.04 and RedHat 7.1, VXLAN interface does not reply to ARP requests for a MAC address that exists in its own ARP table. This issue was fixed in the following newer kernel versions: Ubuntu 16.04 and RedHat 7.3.</p> <p>Workaround: N/A</p> <p>Keywords: ARP, VXLAN</p> <p>Discovered in Release: 4.2-1.0.1.0</p>
1134323	<p>Description: When using kernel versions older than version 4.7 with IOMMU enabled, performance degradations and logical issues (such as soft lockup) might occur upon high load of traffic. This is caused due to the fact that IOMMU IOVA allocations are centralized, requiring many synchronization operations and high locking overhead amongst CPUs.</p> <p>Workaround: Use kernel v4.7 or above, or a backported kernel that includes the following patches:</p> <ul style="list-style-type: none"> • 2aac630429d9 iommu/vt-d: change intel-iommu to use IOVA frame numbers • 9257b4a206fc iommu/iova: introduce per-cpu caching to iova allocation • 22e2f9fa63b0 iommu/vt-d: Use per-cpu IOVA caching <p>Keywords: IOMMU, soft lockup</p> <p>Discovered in Release: 4.2-1.0.1.0</p>
1135738	<p>Description: On 64k page size setups, DMA memory might run out when trying to increase the ring size/number of channels.</p> <p>Workaround: Reduce the ring size/number of channels.</p> <p>Keywords: DMA, 64K page</p> <p>Discovered in Release: 4.2-1.0.1.0</p>
1159650	<p>Description: When configuring VF VST, VLAN-tagged outgoing packets will be dropped in case of ConnectX-4 HCAs. In case of ConnectX-5 HCAs, VLAN-tagged outgoing packets will have another VLAN tag inserted.</p> <p>Workaround: N/A</p> <p>Keywords: VST</p> <p>Discovered in Release: 4.2-1.0.1.0</p>
1157770	<p>Description: On Passthrough/VM machines with relatively old QEMU and libvirt, CMD timeout might occur upon driver load. After timeout, no other commands will be completed and all driver operations will be stuck.</p> <p>Workaround: Upgrade the QEMU and libvirt on the KVM server. Tested with (Ubuntu 16.10) are the following versions:</p> <ul style="list-style-type: none"> • libvirt 2.1.0 • QEMU 2.6.1 <p>Keywords: QEMU</p> <p>Discovered in Release: 4.2-1.0.1.0</p>

Internal Ref. Number	Issue
1147703	Description: Using dm-multipath for High Availability on top of NVMeoF block devices must be done with “directio” path checker.
	Workaround: N/A
	Keywords: NVMeoF
	Discovered in Release: 4.2-1.0.1.0
1152408	Description: RedHat v7.3 PPCLE and v7.4 PPCLE operating systems do not support KVM qemu out of the box. The following error message will appear when attempting to run <i>virt-install</i> to create new VMs: <i>Cant find qemu-kvm packge to install</i>
	Workaround: Acquire the following rpms from the beta version of 7.4ALT to 7.3/7.4 PPCLE (in the same order): <ul style="list-style-type: none"> • qemu-img-.el7a.ppc64le.rpm • qemu-kvm-common-.el7a.ppc64le.rpm • qemu-kvm-.el7a.ppc64le.rpm
	Keywords: Virtualization, PPC, Power8, KVM, RedHat, PPC64LE
	Discovered in Release: 4.2-1.0.1.0
1012719	Description: A soft lockup in the CQ polling flow might occur when running very high stress on the GSI QP (RDMA-CM applications). This is a transient situation from which the driver will later recover.
	Workaround: N/A
	Keywords: RDMA-CM, GSI QP, CQ
	Discovered in Release: 4.2-1.0.1.0
1078630	Description: When working in RoCE LAG over kernel v3.10, a kernel crash might occur when unloading the driver as the Network Manager is running.
	Workaround: Stop the Network Manager before unloading the driver and start it back once the driver unload is complete.
	Keywords: RoCE LAG, network manager
	Discovered in Release: 4.2-1.0.1.0
1149557	Description: When setting VGT+, the maximal number of allowed VLAN IDs presented in the sysfs is 813 (up to the first 813).
	Workaround: N/A
	Keywords: VGT+
	Discovered in Release: 4.2-1.0.1.0

Internal Ref. Number	Issue
995665/1165919	Description: In kernels below v4.13, connection between NVMeoF host and target cannot be established in a hyper-threaded system with more than 1 socket.
	Workaround: On the host side, connect to NVMeoF subsystem using <i>--nr-io-queues <num_queues></i> flag. Note that <i>num_queues</i> must be lower or equal to <i>num_sockets</i> multiplied with <i>num_cores_per_socket</i> .

Internal Ref. Number	Issue
	Keywords: NVMeoF
1039346	<p>Description: Enabling multiple namespaces per subsystem while using NVMeoF target offload is not supported.</p> <p>Workaround: To enable more than one namespace, create a subsystem for each one.</p> <p>Keywords: NVMeoF Target Offload, namespace</p>
1030301	<p>Description: Creating virtual functions on a device that is in LAG mode will destroy the LAG configuration. The bonding device over the Ethernet NICs will continue to work as expected.</p> <p>Workaround: N/A</p> <p>Keywords: LAG, SR-IOV</p>
1047616	<p>Description: When node GUID of a device is set to zero (0000:0000:0000:0000), RDMA_CM user space application may crash.</p> <p>Workaround: Set node GUID to a nonzero value.</p> <p>Keywords: RDMA_CM</p>
1051701	<p>Description: New versions of iproute which support new kernel features may misbehave on old kernels that do not support these new features.</p> <p>Workaround: N/A</p> <p>Keywords: iproute</p>
1007830	<p>Description: When working on Xenserver hypervisor with SR-IOV enabled on it, make sure the following instructions are applied:</p> <ol style="list-style-type: none"> 1. Right after enabling SR-IOV, unbind all driver instances of the virtual functions from their PCI slots. 2. It is not allowed to unbind PF driver instance while having active VFs. <p>Workaround: N/A</p> <p>Keywords: SR-IOV</p>
1005786	<p>Description: When using ConnectX-5 adapter cards, the following error might be printed to dmesg, indicating temporary lack of DMA pages:</p> <pre> mlx5_core ... give_pages:289:(pid x): Y pages alloc time exceeded the max permitted duration mlx5_core ... page_notify_fail:263:(pid x): Page allocation failure notification on func_id(z) sent to fw mlx5_core ... pages_work_handler:471:(pid x): give fail -12" </pre> <p>Example: This might happen when trying to open more than 64 VFs per port.</p> <p>Workaround: N/A</p> <p>Keywords: mlx5_core, DMA</p>
1008066/1009004	<p>Description: Performing some operations on the user end during reboot might cause call trace/panic, due to bugs found in the Linux kernel. For example: Running <code>get_vf_stats</code> (via <code>iptool</code>) during reboot.</p> <p>Workaround: N/A</p> <p>Keywords: mlx5_core, reboot</p>

Internal Ref. Number	Issue
1009488	<p>Description: Mounting MLNX_EN to a path that contains special characters, such as parenthesis or spaces is not supported. For example, when mounting MLNX_EN to “/media/CDROM(vcd)/”, installation will fail and the following error message will be displayed:</p> <pre># cd /media/CDROM(vcd)/ # ./install sh: 1: Syntax error: "(" unexpected</pre> <p>Workaround: N/A</p> <p>Keywords: Installation</p>
982144	<p>Description: When offload traffic sniffer is on, the bandwidth could decrease up to 50%.</p> <p>Workaround: N/A</p> <p>Keywords: Offload Traffic Sniffer</p>
981362	<p>Description: On several OSs, setting a number of TC is not supported via the tc tool.</p> <p>Workaround: Set the number of TC via the /sys/class/net/<interface>/qos/tc_num sysfs file.</p> <p>Keywords: Ethernet, TC</p>
979457	<p>Description: When setting IOMMU=ON, a severe performance degradation may occur due to a bug in IOMMU.</p> <p>Workaround: Make sure the following patches are found in your kernel:</p> <ul style="list-style-type: none"> • iommu/vt-d: Fix PASID table allocation • iommu/vt-d: Fix IOMMU lookup for SR-IOV Virtual Functions <p>Note: These patches are already available in Ubuntu 16.04.02 and 17.04 OSs.</p> <p>Keywords: Performance, IOMMU</p>

2 User Manual

- [Introduction](#)
- [Installation](#)
- [Features Overview and Configuration](#)
- [Troubleshooting](#)
- [Common Abbreviations and Related Documents](#)

2.1 Introduction

This manual is intended for system administrators responsible for the installation, configuration, management and maintenance of the software and hardware of Ethernet adapter cards. It is also intended for application developers.

This document provides information about MLNX_EN Linux driver, and instructions on how to install the driver on ConnectX network adapter solutions supporting the following uplinks to servers:

Uplink/NICs	Driver Name	Uplink Speed
BlueField-2	mlx5	<ul style="list-style-type: none">• InfiniBand: SDR, FDR, EDR, HDR• Ethernet: 1GbE, 10GbE, 25GbE, 40GbE, 50GbE², 100GbE²
BlueField		<ul style="list-style-type: none">• InfiniBand: SDR, QDR, FDR, FDR10, EDR• Ethernet: 1GbE, 10GbE, 25GbE, 40GbE, 50GbE, 100GbE
ConnectX-6 Dx		<ul style="list-style-type: none">• Ethernet: 10GbE, 25GbE, 40GbE, 50GbE², 100GbE², 200GbE²
ConnectX-6 Lx		<ul style="list-style-type: none">• Ethernet: 1GbE, 10GbE, 25GbE, 40GbE, 50GbE²
ConnectX-6		<ul style="list-style-type: none">• InfiniBand: SDR, FDR, EDR, HDR• Ethernet: 10GbE, 25GbE, 40GbE, 50GbE², 100GbE², 200GbE²
ConnectX-5/ConnectX-5 Ex		<ul style="list-style-type: none">• InfiniBand: SDR, QDR, FDR, FDR10, EDR• Ethernet: 1GbE, 10GbE, 25GbE, 40GbE, 50GbE, 100GbE
ConnectX-4 Lx		<ul style="list-style-type: none">• Ethernet: 1GbE, 10GbE, 25GbE, 40GbE, 50GbE
ConnectX-4		<ul style="list-style-type: none">• InfiniBand: SDR, QDR, FDR, FDR10, EDR• Ethernet: 1GbE, 10GbE, 25GbE, 40GbE, 50GbE, 56GbE¹, 100GbE

1. 56GbE is an NVIDIA proprietary link speed and can be achieved while connecting an NVIDIA adapter card to NVIDIA SX10XX switch series or when connecting an NVIDIA adapter card to another NVIDIA adapter card.
2. Supports both NRZ and PAM4 modes.

MLNX_EN driver release exposes the following capabilities:

- Single/Dual port
- Multiple Rx and Tx queues
- Rx steering mode: Receive Core Affinity (RCA)

- MSI-X or INTx
- Adaptive interrupt moderation
- HW Tx/Rx checksum calculation
- Large Send Offload (i.e., TCP Segmentation Offload)
- Large Receive Offload
- Multi-core NAPI support
- VLAN Tx/Rx acceleration (HW VLAN stripping/insertion)
- Ehtool support
- Net device statistics
- SR-IOV support
- Flow steering
- Ethernet Time Stamping

2.1.1 Package Contents

2.1.1.1 Package Images

MLNX_EN is provided as an ISO image or as a tarball per Linux distribution and CPU architecture that includes source code and binary RPMs, firmware and utilities. The ISO image contains an installation script (called install) that performs the necessary steps to accomplish the following:

- Discover the currently installed kernel
- Uninstall any previously installed MLNX_OFED/MLNX_EN packages
- Install the MLNX_EN binary RPMs (if they are available for the current kernel)
- Identify the currently installed HCAs and perform the required firmware updates

2.1.1.2 Software Components

MLNX_EN contains the following software components:

Components	Description
mlx5 driver	mlx5 is the low level driver implementation for the ConnectX-4 adapters. ConnectX-4 operates as a VPI adapter.
mlx5_core	Acts as a library of common functions (e.g. initializing the device after reset) required by the ConnectX-4 adapter cards.
mlx4 driver	mlx4 is the low level driver implementation for the ConnectX adapters. The ConnectX can operate as an InfiniBand adapter and as an Ethernet NIC. To accommodate the two flavors, the driver is split into modules: mlx4_core, mlx4_en, and mlx4_ib. Note: mlx4_ib is not part of this package.
mstflint	An application to burn a firmware binary image.
Software modules	Source code for all software modules (for use under conditions mentioned in the modules' LICENSE files)

2.1.1.3 Firmware

The ISO image includes the following firmware item:

- Firmware images (.bin format wrapped in the mlxfwmanager tool) for ConnectX-4 and above network adapters

2.1.1.4 Directory Structure

The tarball image of MLNX_EN contains the following files and directories:

- install—the MLNX_EN installation script
- uninstall.sh—the MLNX_EN un-installation script
- RPMS/—directory of binary RPMs for a specific CPU architecture
- src/—directory of the OFED source tarball
- mlnx_add_kernel_support.sh—a script required to rebuild MLNX_EN for customized kernel version on supported Linux distribution

2.1.2 Module Parameters

2.1.2.1 mlx5_core Module Parameters

The mlx5_core module supports a single parameter used to select the profile which defines the number of resources supported.

<i>prof_sel</i>	The parameter name for selecting the profile. The supported values for profiles are: <ul style="list-style-type: none">• 0—for medium resources, medium performance• 1—for low resources• 2—for high performance (int) (default)
guids	charp
node_guid	guids configuration. This module parameter will be obsolete!
debug_mask	debug_mask: 1 = dump cmd data, 2 = dump cmd exec time, 3 = both. Default=0 (uint)
probe_vf	probe VFs or not, 0 = not probe, 1 = probe. Default = 1 (bool)
num_of_groups	Controls the number of large groups in the FDB flow table. Default=4; Range=1-1024

2.1.3 Devlink Parameters

The following parameters, supported in mlx4 driver only, can be changed using the Devlink user interface:

Parameter	Description	Parameter Type
internal_error_reset	Enables resetting the device on internal errors	Generic
max_macs	Max number of MACs per ETH port	Generic
region_snapshot_enable	Enables capturing region snapshots	Generic
enable_64b_cqe_eqe	Enables 64 byte CQEs/EQEs when supported by FW	Driver-specific
enable_4k_uar	Enables using 4K UAR	Driver-specific

2.2 Installation

This chapter describes how to install and test the NVIDIA OFED for Linux package on a single host machine with NVIDIA InfiniBand and/or Ethernet adapter hardware installed.

The chapter contains the following sections:

- [Software Dependencies](#)
- [Downloading the Drivers](#)
- [Installing MLNX_EN](#)
- [Uninstall](#)
- [Updating Firmware After Installation](#)
- [Ethernet Driver Usage and Configuration](#)
- [Performance Tuning](#)

2.2.1 Software Dependencies

MLNX_EN driver cannot coexist with OFED software on the same device. Therefore, when installing MLNX_EN, all OFED packages should be removed (by running the `install` script).

2.2.2 Downloading the Drivers

1. Verify that the system has a NVIDIA network adapter (HCA/NIC) installed.

The following example shows a system with an installed NVIDIA HCA:

```
# lspci -v | grep Mellanox
86:00.0 Network controller [0207]: Mellanox Technologies MT27620 Family
Subsystem: Mellanox Technologies Device 0014
86:00.1 Network controller [0207]: Mellanox Technologies MT27620 Family
Subsystem: Mellanox Technologies Device 0014
```

Note: For ConnectX-5 Socket Direct adapters, use `ibdev2netdev` to display the installed card and the mapping of logical ports to physical ports. Example:

```
[root@gen-1-vrt-203 ~]# ibdev2netdev -v | grep -i MCX556M-ECAT-S25
0000:84:00.0 mlx5_10 (MT4119 - MCX556M-ECAT-S25SN) CX556M - ConnectX-5 QSFP28 fw 16.22.0228 port 1 (DOWN)
==> p2p1 (Down)
0000:84:00.1 mlx5_11 (MT4119 - MCX556M-ECAT-S25SN) CX556M - ConnectX-5 QSFP28 fw 16.22.0228 port 1 (DOWN)
==> p2p2 (Down)
0000:05:00.0 mlx5_2 (MT4119 - MCX556M-ECAT-S25SN) CX556M - ConnectX-5 QSFP28 fw 16.22.0228 port 1 (DOWN)
==> p5p1 (Down)
0000:05:00.1 mlx5_3 (MT4119 - MCX556M-ECAT-S25SN) CX556M - ConnectX-5 QSFP28 fw 16.22.0228 port 1 (DOWN)
==> p5p2 (Down)
```

- Each PCI card of ConnectX-5 Socket Direct has a different PCI address. In the output example above, the first two rows indicate that one card is installed in a PCI slot with PCI Bus address 84 (hexadecimal), and PCI Device Number 00, and PCI Function Number 0 and 1. RoCE assigned `mlx5_10` as the logical port, which is the same as netdevice `p2p1`, and both are mapped to physical port of PCI function `0000:84:00.0`.
- RoCE logical port `mlx5_2` of the second PCI card (PCI Bus address 05) and netdevice `p5p1` are mapped to physical port of PCI function `0000:05:00.0`, which is the same physical port of PCI function `0000:84:00.0`.
MT4119 is the PCI Device ID of the ConnectX-5 adapters family.

For more details, please refer to ConnectX-5 Socket Direct Hardware User Manual, available at nvidia.com/en-us/networking/.

2. Download the ISO image to your host.
The image name has the format `mlnx-en-<ver>-<OS label>-<CPU arch>.iso`. It can also be downloaded from nvidia.com/en-us/networking/ → Products → Software → Ethernet Drivers.
3. Use the `md5sum` utility to confirm the file integrity of your ISO image. Run the following command and compare the result to the value provided on the download page.

2.2.3 Installing MLNX_EN

2.2.3.1 Installation Script

The `install` installation script performs the following:

- Discovers the currently installed kernel
- Uninstalls any previously installed MLNX_EN package
- Installs the MLNX_EN binary (if they are available for the current kernel)
- Identifies the currently installed Ethernet network adapters and automatically upgrades the firmware

If the driver detects unsupported cards on your system, it will abort the installation procedure. To avoid this, make sure to add `--skip-unsupported-devices-check` flag during installation.

2.2.3.2 Installation Modes

`mlnx_en` installer supports 2 modes of installation. The install script selects the mode of driver installation depending on the running OS/kernel version.

- Kernel Module Packaging (KMP) mode, where the source rpm is rebuilt for each installed flavor of the kernel. This mode is used for RedHat and SUSE distributions.

- Non KMP installation mode, where the sources are rebuilt with the running kernel. This mode is used for vanilla kernels.
 - By default, the package will install drivers supporting Ethernet only. In addition, the package will include the following new installation options:
 - Full VMA support which can be installed using the installation option “-vma”.
 - Infrastructure to run DPDK using the installation option “-dpdk”.
- Notes:
- DPDK itself is not included in the package. Users would still need to install DPDK separately after the MLNX_EN installation is completed.
 - Installing VMA or DPDK infrastructure will allow users to run RoCE.

Installation Results

- For Ethernet only installation mode:
 - The kernel modules are installed under:
 - /lib/modules/`uname -r`/updates on SLES and Fedora Distributions
 - /lib/modules/`uname -r`/extra/mlnx-en on RHEL and other RedHat like Distributions
 - /lib/modules/`uname -r`/updates/dkms/ on Ubuntu
 - The kernel module sources are placed under:


```
/usr/src/mlnx-en-<ver>/
```
- For VPI installation mode:
 - The kernel modules are installed under:
 - /lib/modules/`uname -r`/updates on SLES and Fedora Distributions
 - /lib/modules/`uname -r`/extra/mlnx-ofa_kernel on RHEL and other RedHat like Distributions
 - /lib/modules/`uname -r`/updates/dkms/ on Ubuntu
 - The kernel module sources are placed under:


```
/usr/src/ofa_kernel-<ver>/
```

2.2.3.3 Installation Procedure

This section describes the installation procedure of MLNX_EN on NVIDIA adapter cards.

1. Log into the installation machine as root.
2. Mount the ISO image on your machine.

```
host1# mount -o ro,loop mlnx-en-<ver>-<OS label>-<CPU arch>.iso /mnt
```

3. Run the installation script.

```
/mnt/install
```

4. Case A: If the installation script has performed a firmware update on your network adapter, you need to either restart the driver or reboot your system before the firmware update can take effect. Refer to the table below to find the appropriate action for your specific card.

Action \ Adapter	Driver Restart	Standard Reboot (Soft Reset)	Cold Reboot (Hard Reset)
------------------	----------------	------------------------------	--------------------------

Standard ConnectX-4/ ConnectX-4 Lx or higher	-	+	-
Adapters with Multi- Host Support	-	-	+
Socket Direct Cards	-	-	+

Case B: If the installations script has not performed a firmware upgrade on your network adapter, restart the driver by running: `# /etc/init.d/mlnx-en.d restart`

The result is a new net-device appearing in the 'ifconfig -a' output.

2.2.3.4 Additional Installation Procedures

2.2.3.4.1 Installing MLNX_EN Using YUM

This type of installation is applicable to RedHat/OL, Fedora, XenServer operating systems.

2.2.3.4.1.1 Setting up MLNX_EN YUM Repository

The package consists of two folders that can be set up as a repository:

- “RPMS_ETH” - provides the Ethernet only installation mode
- “RPMS” - provides the RDMA support installation mode

1. Log into the installation machine as `root`.
2. Mount the ISO image on your machine and copy its content to a shared location in your network.

```
# mount -o ro,loop mlnx-en-<ver>-<OS label>-<CPU arch>.iso /mnt
```

You can download the image from nvidia.com/en-us/networking/.com → Products → Software → Ethernet Drivers.

3. Download and install the NVIDIA GPG-KEY:

The key can be downloaded via the following link: <http://www.mellanox.com/downloads/ofed/RPM-GPG-KEY-Mellanox>

Example:

```
# wget http://www.mellanox.com/downloads/ofed/RPM-GPG-KEY-Mellanox
--2014-04-20 13:52:30-- http://www.mellanox.com/downloads/ofed/RPM-GPG-KEY-Mellanox
Resolving www.mellanox.com... 72.3.194.0
Connecting to www.mellanox.com[72.3.194.0]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1354 (1.3K) [text/plain]
Saving to: ?RPM-GPG-KEY-Mellanox?
100%[=====] 1,354 --.-K/s in 0s
2014-04-20 13:52:30 (247 MB/s) - ?RPM-GPG-KEY-Mellanox? saved [1354/1354]
```

4. Install the key.

Example:

```
# sudo rpm --import RPM-GPG-KEY-Mellanox
warning: rpmts_HdrFromFdno: Header V3 DSA/SHA1 Signature, key ID 6224c050: NOKEY
Importing GPG key 0x6224C050:
Userid: "Mellanox Technologies (Mellanox Technologies - Signing Key v2) <support@mellanox.com>"
```

```
From : /repos/MLNX_EN/RPM-GPG-KEY-Mellanox
Is this ok [y/N]:
```

5. Check that the key was successfully imported.

Example:

```
# rpm -q gpg-pubkey --qf '%{NAME}-%{VERSION}-%{RELEASE}\t%{SUMMARY}\n' | grep Mellanox
gpg-pubkey-a9e4b643-520791ba gpg(Mellanox Technologies <support@mellanox.com>)
Rev 3.30
Mellanox Technologies 45
```

6. Create a YUM repository configuration file called “/etc/yum.repos.d/mlnx_en.repo” with the following content:

```
[mlnx_en]
name=MLNX_EN Repository
baseurl=file:///<path to extracted MLNX_EN package>/<RPMS FOLDER NAME>
enabled=1
gpgkey=file:///<path to the downloaded key RPM-GPG-KEY-Mellanox>
gpgcheck=1
```

Replace <RPMS FOLDER NAME> with “RPMS_ETH” or “RPMS” depending on the desired installation mode (Ethernet only or RDMA).

7. Check that the repository was successfully added.

```
# yum repolist
Loaded plugins: product-id, security, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use subscription-manager to register.
repo id                                repo name
status                                  MLNX_EN Repository
mlnx_en
```

2.2.3.4.1.2 Installing MLNX_EN Using the YUM Tool

After setting up the YUM repository for MLNX_EN package, install one of the following metadata packages:

- In case you set up the “RPMS_ETH” folder as the repository (for Ethernet only mode), install:

```
# yum install mlnx-en-eth-only
```

- In case you set up the “RPMS” folder as the repository (for RDMA mode), install either:

```
# yum install mlnx-en-vma
```

Or

```
# yum install mlnx-en-dpdk
```

Please note the following:

MLNX_EN provides kernel module RPM packages with KMP support for RHEL and SLES. For other operating systems, kernel module RPM packages are provided only for the operating system’s default kernel. In this case, the group RPM packages have the supported kernel version in their packages name.

If you have an operating systems different than RHEL or SLES, or you have installed a kernel that is not supported by default in MLNX_EN, you can use the `mlnx_add_kernel_support.sh` script to build MLNX_EN for your kernel.

The script will automatically build the matching group RPM packages for your kernel so that you can still install MLNX_EN via YUM.

Please note that the resulting MLNX_EN repository will contain unsigned RPMs. Therefore, you should set `'gpgcheck=0'` in the repository configuration file.

Installing MLNX_EN using the YUM tool does not automatically update the firmware.

To update the firmware to the version included in MLNX_EN package, you can either:

1. Run:

```
# yum install mlnx-fw-updater
```

OR

2. Update the firmware to the latest version available on NVIDIA website as described in [Updating Firmware After Installation](#) section.

2.2.3.4.2 Installing MLNX_EN Using apt-get

This type of installation is applicable to Debian and Ubuntu operating systems.

2.2.3.4.2.1 Setting up MLNX_EN apt-get Repository

The package consists of two folders that can be set up as a repository:

- “DEBS_ETH” - provides the Ethernet only installation mode.
- “RPMS” - provides the RDMA support installation mode.

1. Log into the installation machine as root.
2. Extract the MLNX_EN package on a shared location in your network.
You can download it from [nvidia.com/en-us/networking/.com](https://www.nvidia.com/en-us/networking/.com) → Products → Software → Ethernet Drivers.
3. Create an apt-get repository configuration file called `"/etc/apt/sources.list.d/mlnx_en.list"` with the following content:

```
deb file:./<path to extracted MLNX_EN package>/<DEBS FOLDER NAME> ./
```

Replace `<DEBS FOLDER NAME>` with “DEBS_ETH” or “DEBS” depending on the desired installation mode (Ethernet only or RDMA).

4. Download and install the NVIDIA GPG-KEY.

Example:

```
# wget -qO - http://www.mellanox.com/downloads/ofed/RPM-GPG-KEY-Mellanox | sudo apt-key add -
```

5. Verify that the key was successfully imported.

Example:


```
# apt-key list
pub 1024D/A9E4B643 2013-08-11
uid Mellanox Technologies <support@mellanox.com>
sub 1024g/09FCC269 2013-08-11
```

6. Update the apt-get cache.

```
# sudo apt-get update
```

2.2.3.4.2.2 Installing MLNX_EN Using the apt-get Tool

After setting up the apt-get repository for MLNX_EN package, install one of the following metadata packages:

- In case you set up the “DEBS_ETH” folder as the repository (for Ethernet only mode), install:

```
# apt-get install mlnx-en-eth-only
```

- In case you set up the “DEBS” folder as the repository (for RDMA mode), install either:

```
# apt-get install mlnx-en-vma
```

OR

```
# apt-get install mlnx-en-dpdk
```

Installing MLNX_EN using the apt-get tool does not automatically update the firmware. To update the firmware to the version included in MLNX_EN package, you can either:

1. Run:

```
# apt-get install mlnx-fw-updater
```

Or

2. Update the firmware to the latest version available on NVIDIA website as described in [Updating Firmware After Installation](#) section.

2.2.3.4.2.3 Installation Using Repositories

NVIDIA Legacy Libraries can also be installed using the operating system's standard package manager (yum, apt-get, etc.).

- For RPM based operating systems, follow the steps in “[Installing MLNX_EN Using YUM](#)” section, using the directory “MLNX_LIBS” instead of “UPSTREAM_LIBS” when creating the “/etc/yum.repos.d/mlnx_ofed.repo” file.
- For Debian based operating systems, follow the steps in “[Installing MLNX_EN Using apt-get](#)” section using the directory “MLNX_LIBS” instead of “UPSTREAM_LIBS” when creating the “/etc/apt/sources.list.d/mlnx_ofed.list” file.

Finally, for both RPM and Debian based OSs, install the new metadata package called “mlnx-ofed-dpdk-upstream-libs”, which will install both the user space and kernel packages.

If you wish to install only the user space packages, make sure to install the metadata package called “mlnx-ofed-dpdk-upstream-libs-user-only”.

2.2.4 Uninstall

2.2.4.1 Uninstalling MLNX_EN Using the YUM and apt-get Tools

Use the script `/usr/sbin/mlnx_en_uninstall.sh` to uninstall MLNX_EN package.

2.2.5 Updating Firmware After Installation

The firmware can be updated using one of the following methods.

2.2.5.1 Updating the Device Online

To update the device online on the machine from the NVIDIA site, use the following command line:

```
mlxfwmanager --online -u -d <device>
```

Example:

```
# mlxfwmanager --online -u -d 0000:81:00.0
Querying Mellanox devices firmware ...

Device #1:
-----
Device Type:      ConnectX6DX
Part Number:     MCX623106AN-CDA_Ax
Description:     ConnectX-6 Dx EN adapter card; 100GbE; Dual-port QSFP56; PCIe 4.0/3.0 x16;
PSID:           MT_0000000359
PCI Device Name: 0000:81:00.0
Base GUID:      1c34da030080284a
Base MAC:       1c34da80284a
Versions:
  FW           22.28.1034  22.28.1002
  PXE          3.6.0101   3.6.0101
  UEFI         14.21.0016   14.21.0016

Status:         Update required
-----
Found 1 device(s) requiring firmware update. Please use -u flag to perform the update.
```

2.2.5.2 Updating the Device Manually

When running the `install` script with the `--without-fw-update` option or using an OEM card that you now wish to (manually) update firmware on your adapter card(s), perform the steps below. The following steps are also appropriate to burn newer firmware that was downloaded from the website (i.e., [nvidia.com/en-us/networking/](https://www.nvidia.com/en-us/networking/) → Support → Support → [Firmware Download](#)).

1. Get the device's PSID.

```
mlxfwmanager_pci | grep PSID
PSID:             MT_1210110019
```

2. Download the firmware BIN file from the website or the OEM website.
3. Burn the firmware.

```
mlxfwmanager_pci -i <fw_file.bin>
```

4. Reboot the device once the firmware burning is completed.

2.2.5.3 Updating the Device Firmware Automatically Upon System Boot

Firmware can be automatically updated upon system boot.

The firmware update package (mlnx-fw-updater) is installed in the “/opt/mellanox/mlnx-fw-updater” folder, and the openibd service script can invoke the firmware update process if requested on boot.

If the firmware is updated, the following message will be printed to the system’s standard logging file:

```
fw_updater: Firmware was updated. Please reboot your system for the changes to take effect.
```

Otherwise, the following message will be printed:

```
fw_updater: Didn't detect new devices with old firmware.
```

Please note that this feature is disabled by default. To enable the automatic firmware update upon system boot, set the following parameter to “yes” “RUN_FW_UPDATER_ONBOOT=yes” in the openibd service configuration file “/etc/infiniband/openib.conf”.

You can opt to exclude a list of devices from the automatic firmware update procedure. To do so, edit the configurations file “/opt/mellanox/mlnx-fw-updater/mlnx-fw-updater.conf” and provide a comma separated list of PCI devices to exclude from the firmware update.

Example:

```
MLNX_EXCLUDE_DEVICES="00:05.0,00:07.0"
```

2.2.6 Ethernet Driver Usage and Configuration

➤ *To assign an IP address to the interface, run:*

```
#> ifconfig eth<x> <ip>
```

Note: 'x' is the OS assigned interface number.

➤ *To check driver and device information:*

```
#> ethtool -i eth<x>
```

Example:

```
#> ethtool -i eth2
driver: mlx4_en
version: 2.1.8 (Oct 06 2013)
firmware-version: 2.30.3110
bus-info: 0000:1a:00.0
```

➤ *To query stateless offload status:*

```
#> ethtool -k eth<x>
```

➤ *To set stateless offload status:*

```
#> ethtool -K eth<x> [rx on|off] [tx on|off] [sg on|off] [tso on|off] [lro on|off]
```

➤ *To query interrupt coalescing settings:*

```
#> ethtool -c eth<x>
```

➤ *To enable/disable adaptive interrupt moderation:*

```
#>ethtool -C eth<x> adaptive-rx on|off
```

By default, the driver uses adaptive interrupt moderation for the receive path, which adjusts the moderation time to the traffic pattern.

➤ *To set the values for packet rate limits and for moderation time high and low:*

```
#> ethtool -C eth<x> [pkt-rate-low N] [pkt-rate-high N] [rx-usecs-low N] [rx-usecs-high N]
```

Above an upper limit of packet rate, adaptive moderation will set the moderation time to its highest value. Below a lower limit of packet rate, the moderation time will be set to its lowest value.

➤ *To set interrupt coalescing settings when adaptive moderation is disabled:*

```
#> ethtool -C eth<x> [rx-usecs N] [rx-frames N]
```

usec settings correspond to the time to wait after the *last* packet is sent/received before triggering an interrupt.

➤ *To query ring size values:*

```
#> ethtool -g eth<x>
```

➤ *To modify rings size:*

```
#> ethtool -G eth<x> [rx <N>] [tx <N>]
```

➤ *To obtain additional device statistics:*

```
#> ethtool -S eth<x>
```

The driver defaults to the following parameters:

- Both ports are activated (i.e., a net device is created for each port)
- The number of Rx rings for each port is the nearest power of 2 of number of cpu cores, limited by 16.

- LRO is enabled with 32 concurrent sessions per Rx ring

Some of these values can be changed using module parameters, which can be displayed by running:

```
#> modinfo mlx5_en
```

To set non-default values to module parameters, add to the `/etc/modprobe.conf` file:

```
"options mlx5_en <param_name>=<value> <param_name>=<value> ..."
```

Values of all parameters can be observed in `/sys/module/mlx5_en/parameters/`.

2.2.7 Performance Tuning

Depending on the application of the user's system, it may be necessary to modify the default configuration of network adapters based on the ConnectX® adapters. In case tuning is required, please refer to the [Performance Tuning for NVIDIA Adapters](#) Community post.

2.3 Features Overview and Configuration

Unable to render include or excerpt-include. Could not retrieve page.

The chapter contains the following sections:

- [Ethernet Network](#)
- [Virtualization](#)
- [Resiliency](#)
- [Docker Containers](#)
- [Fast Driver Unload](#)
- [OVS Offload Using ASAP² Direct](#)

2.3.1 Ethernet Network

The chapter contains the following sections:

- [Ethernet Interface](#)
- [Quality of Service \(QoS\)](#)
- [Ethtool](#)
- [Checksum Offload](#)
- [Ignore Frame Check Sequence \(FCS\) Errors](#)
- [RDMA over Converged Ethernet \(RoCE\)](#)
- [Flow Control](#)
- [Explicit Congestion Notification \(ECN\)](#)
- [RSS Support](#)
- [Time-Stamping](#)
- [Flow Steering](#)

- [Wake-on-LAN \(WoL\)](#)
- [Hardware Accelerated 802.1ad VLAN \(Q-in-Q Tunneling\)](#)
- [VLAN Stripping in Linux Verbs](#)
- [Offloaded Traffic Sniffer](#)
- [Dump Configuration](#)
- [Local Loopback Disable](#)
- [Kernel Transport Layer Security \(kTLS\) Offloads](#)
- [IPsec Crypto Offload](#)
- [MACsec Full Offload](#)

2.3.1.1 Ethernet Interface

2.3.1.1.1 Counters

Counters are used to provide information about how well an operating system, an application, a service, or a driver is performing. The counter data help determine system bottlenecks and fine-tune the system and application performance. The operating system, network, and devices provide counter data that an application can consume to provide users with a graphical view of how well the system is performing.

The counter index is a Queue Pair (QP) attribute given in the QP context. Multiple QPs may be associated with the same counter set. If multiple QPs share the same counter, the counter value will represent the cumulative total.

2.3.1.1.1.1 RoCE Counters

- RoCE counters are available only through sysfs located under:
 - `# /sys/class/infiniband/<device>/ports/*/counters/`
 - `# /sys/class/infiniband/<device>/ports/*/hw_counters/`

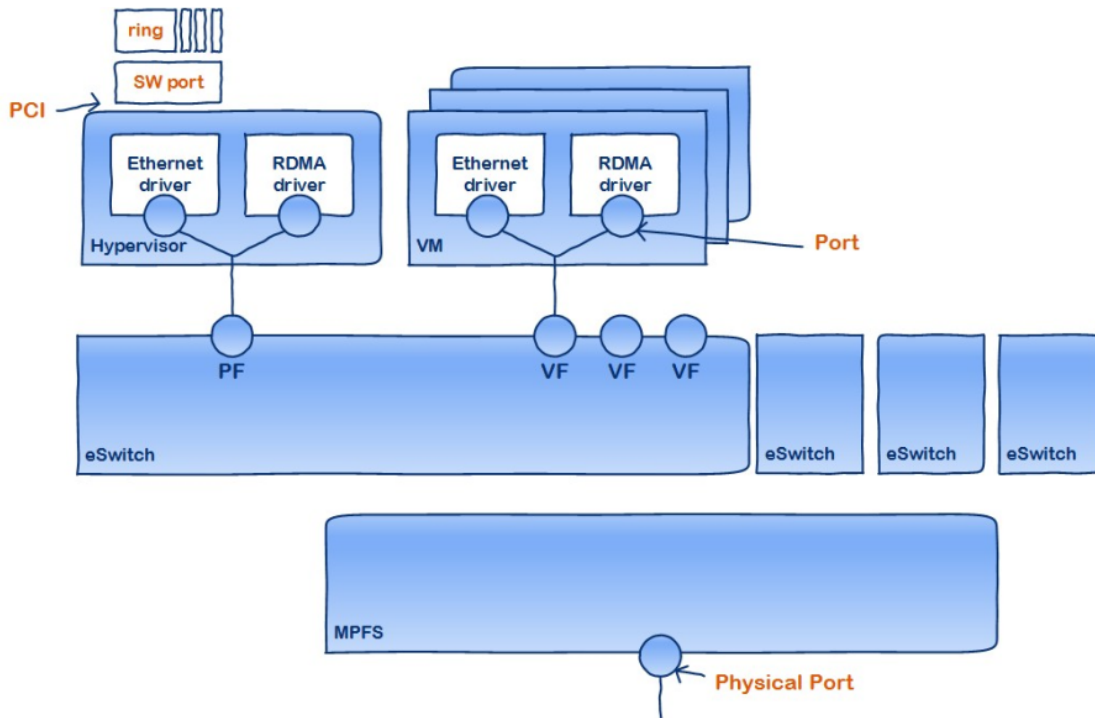
For mlx5 port and RoCE counters, refer to the [Understanding mlx5 Linux Counters](#) Community post.

2.3.1.1.1.2 SR-IOV Counters

Physical Function can also read Virtual Functions' port counters through sysfs located under `# /sys/class/net/<interface_name>/device/sriov/<index>/stats/`

2.3.1.1.1.3 ethtool Counters

The ethtool counters are counted in different places, according to which they are divided into groups. Each counters group may also have different counter types.



For the full list of supported ethtool counters, refer to the [Understanding mlx5 Linux Counters](#) Community post.

2.3.1.1.2 Persistent Naming

To avoid network interface renaming after boot or driver restart, set the desired constant interface name in the "/etc/udev/rules.d/70-persistent-net.rules" file.

- Example for Ethernet interfaces:

```
PCI device 15b3:1019 (mlx5_core)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="00:02:c9:fa:c3:50", ATTR{dev_id}=="0x0",
ATTR{type}=="1", KERNEL=="eth*", NAME="eth1"
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="00:02:c9:fa:c3:51", ATTR{dev_id}=="0x0",
ATTR{type}=="1", KERNEL=="eth*", NAME="eth2"
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="00:02:c9:e9:56:a1", ATTR{dev_id}=="0x0",
ATTR{type}=="1", KERNEL=="eth*", NAME="eth3"
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="00:02:c9:e9:56:a2", ATTR{dev_id}=="0x0",
ATTR{type}=="1", KERNEL=="eth*", NAME="eth4"
```

- Example for IPoB interfaces:

```
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{dev_id}=="0x0", ATTR{type}=="32", NAME="ib0"
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{dev_id}=="0x1", ATTR{type}=="32", NAME="ib1"
```

2.3.1.1.3 Interrupt Request (IRQ) Naming

Once IRQs are allocated by the driver, they are named `mlx5_comp<x>@pci:<pci_addr>`. The IRQs corresponding to the channels in use are renamed to `<interface>-<x>`, while the rest maintain their default name.

The mlx5_core driver allocates all IRQs during loading time to support the maximum possible number of channels. Once the driver is up, no further IRQs are freed or allocated. Changing the number of working channels does not re-allocate or free the IRQs.

The following example demonstrates how reducing the number of channels affects the IRQs names.

```

$ ethtool -l ens1
Channel parameters for ens1:
Pre-set maximums:
RX:          0
TX:          0
Other:       0
Combined:    12

Current hardware settings:
RX:          0
TX:          0
Other:       0
Combined:    12

$ cat /proc/interrupts
98:          0          0          0          0          0          7935          0          0          0
0:           0 IR-PCI-MSI-edge  mlx5_async@pci:0000:81:00.0
99:          0          0          0          0          0          1          0          0          0
0:           0 IR-PCI-MSI-edge  ens1-0
100:         0          0          0          0          0          1          0          0          0
0:           0 IR-PCI-MSI-edge  ens1-1
101:         0          0          0          0          0          1          0          0          0
0:           0 IR-PCI-MSI-edge  ens1-2
102:         0          0          0          0          0          1          0          0          0
0:           0 IR-PCI-MSI-edge  ens1-3
103:         0          0          0          0          0          1          0          0          0
0:           0 IR-PCI-MSI-edge  ens1-4
104:         0          0          0          0          0          1          0          0          0
0:           0 IR-PCI-MSI-edge  ens1-5
105:         0          0          0          0          0          1          0          0          0
0:           0 IR-PCI-MSI-edge  ens1-6
106:         0          0          0          0          0          1          0          0          0
0:           0 IR-PCI-MSI-edge  ens1-7
107:         0          0          0          0          0          1          0          0          0
0:           0 IR-PCI-MSI-edge  ens1-8
108:         0          0          0          0          0          1          0          0          0
0:           0 IR-PCI-MSI-edge  ens1-9
109:         0          0          0          0          0          1          0          0          0
0:           0 IR-PCI-MSI-edge  ens1-10
110:        0          0          0          0          0          1          0          0          0
0:           0 IR-PCI-MSI-edge  ens1-11

$ ethtool -L ens1 combined 4
$ ethtool -l ens1
Channel parameters for ens1:
...
Current hardware settings:
RX:          0
TX:          0
Other:       0
Combined:    4

$ cat /proc/interrupts
98:          0          0          0          0          0          8455          0          0          0
0:           0 IR-PCI-MSI-edge  mlx5_async@pci:0000:81:00.0
99:          0          0          0          0          0          1          2          0          0
0:           0 IR-PCI-MSI-edge  ens1-0
100:         0          0          0          0          0          1          0          2          0
0:           0 IR-PCI-MSI-edge  ens1-1
101:         0          0          0          0          0          1          0          0          2
0:           0 IR-PCI-MSI-edge  ens1-2
102:         0          0          0          0          0          1          0          0          0
2:          0 IR-PCI-MSI-edge  ens1-3
103:         0          0          0          0          0          1          0          0          0
0:           1 IR-PCI-MSI-edge  mlx5_comp4@pci:0000:81:00.0
104:         0          0          0          0          0          2          0          0          0
0:           0 IR-PCI-MSI-edge  mlx5_comp5@pci:0000:81:00.0
105:         0          0          0          0          0          1          1          0          0
0:           0 IR-PCI-MSI-edge  mlx5_comp6@pci:0000:81:00.0
106:         0          0          0          0          0          1          0          1          0
0:           0 IR-PCI-MSI-edge  mlx5_comp7@pci:0000:81:00.0
107:         0          0          0          0          0          1          0          0          1
0:           0 IR-PCI-MSI-edge  mlx5_comp8@pci:0000:81:00.0
108:         0          0          0          0          0          1          0          0          0
1:           0 IR-PCI-MSI-edge  mlx5_comp9@pci:0000:81:00.0
109:         0          0          0          0          0          1          0          0          0
0:           1 IR-PCI-MSI-edge  mlx5_comp10@pci:0000:81:00.0
110:        0          0          0          0          0          2          0          0          0
0:           0 IR-PCI-MSI-edge  mlx5_comp11@pci:0000:81:00.0

```


2.3.1.2 Quality of Service (QoS)

Quality of Service (QoS) is a mechanism of assigning a priority to a network flow (socket, rdma_cm connection) and manage its guarantees, limitations and its priority over other flows. This is accomplished by mapping the user's priority to a hardware TC (traffic class) through a 2/3 stage process. The TC is assigned with the QoS attributes and the different flows behave accordingly.

2.3.1.2.1 Mapping Traffic to Traffic Classes

Mapping traffic to TCs consists of several actions which are user controllable, some controlled by the application itself and others by the system/network administrators.

The following is the general mapping traffic to Traffic Classes flow:

1. The application sets the required Type of Service (ToS).
2. The ToS is translated into a Socket Priority (sk_prio).
3. The sk_prio is mapped to a User Priority (UP) by the system administrator (some applications set sk_prio directly).
4. The UP is mapped to TC by the network/system administrator.
5. TCs hold the actual QoS parameters

QoS can be applied on the following types of traffic. However, the general QoS flow may vary among them:

- Plain Ethernet - Applications use regular inet sockets and the traffic passes via the kernel Ethernet driver
- RoCE - Applications use the RDMA API to transmit using Queue Pairs (QPs)
- Raw Ethernet QP - Application use VERBs API to transmit using a Raw Ethernet QP

2.3.1.2.2 Plain Ethernet Quality of Service Mapping

Applications use regular inet sockets and the traffic passes via the kernel Ethernet driver. The following is the Plain Ethernet QoS mapping flow:

1. The application sets the ToS of the socket using setsockopt (IP_TOS, value).
2. ToS is translated into the sk_prio using a fixed translation:

```
TOS 0 <=> sk_prio 0
TOS 8 <=> sk_prio 2
TOS 24 <=> sk_prio 4
TOS 16 <=> sk_prio 6
```

3. The Socket Priority is mapped to the UP in the following conditions:
 - a. If the underlying device is a VLAN device, egress_map is used controlled by the vconfig command. This is per VLAN mapping.
 - b. If the underlying device is not a VLAN device, the mapping is done in the driver.
4. The UP is mapped to the TC as configured by the mlnx_qos tool or by the lldpad daemon if DCBX is used.

Socket applications can use setsockopt (SK_PRIO, value) to directly set the sk_prio of the socket. In this case, the ToS to sk_prio fixed mapping is not needed. This allows the application and the administrator to utilize more than the 4 values possible via ToS.

In the case of a VLAN interface, the UP obtained according to the above mapping is also used in the VLAN tag of the traffic.

2.3.1.2.3 RoCE Quality of Service Mapping

Applications use RDMA-CM API to create and use QPs. The following is the RoCE QoS mapping flow:

1. The application sets the ToS of the QP using the `rdma_set_option(option(RDMA_OPTION_ID_TOS, value))`.
2. ToS is translated into the Socket Priority (`sk_prio`) using a fixed translation:

```
TOS 0 <=> sk_prio 0
TOS 8 <=> sk_prio 2
TOS 24 <=> sk_prio 4
TOS 16 <=> sk_prio 6
```

3. The Socket Priority is mapped to the User Priority (UP) using the `tc` command.
 - In the case of a VLAN device where the parent real device is used for the purpose of this mapping
 - If the underlying device is a VLAN device, and the parent real device was not used for the mapping, the VLAN device's `egress_map` is used
4. UP is mapped to the TC as configured by the `mlnx_qos` tool or by the `lldpad` daemon if DCBX is used.

With RoCE, there can only be 4 predefined ToS values for the purpose of QoS mapping.

2.3.1.2.4 Map Priorities with `set_egress_map`

For RoCE old kernels that do not support `set_egress_map`, use the `tc_wrap` script to map between `sk_prio` and UP. Use `tc_wrap` with option `-u`. For example:

```
tc_wrap -i <ethX> -u <skprio2up mapping>
```

2.3.1.2.5 Quality of Service Properties

The different QoS properties that can be assigned to a TC are:

- [Strict Priority](#)
- [Enhanced Transmission Selection \(ETS\)](#)
- [Rate Limit](#)
- [Trust State](#)
- [Receive Buffer](#)
- [DCBX Control Mode](#)

2.3.1.2.5.1 Strict Priority

When setting a TC's transmission algorithm to be 'strict', then this TC has absolute (strict) priority over other TC strict priorities coming before it (as determined by the TC number: TC 7 is the highest priority, TC 0 is lowest). It also has an absolute priority over nonstrict TCs (ETS).

This property needs to be used with care, as it may easily cause starvation of other TCs.

A higher strict priority TC is always given the first chance to transmit. Only if the highest strict priority TC has nothing more to transmit, will the next highest TC be considered.

Nonstrict priority TCs will be considered last to transmit.

This property is extremely useful for low latency low bandwidth traffic that needs to get immediate service when it exists, but is not of high volume to starve other transmitters in the system.

2.3.1.2.5.2 Enhanced Transmission Selection (ETS)

Enhanced Transmission Selection standard (ETS) exploits the time periods in which the offered load of a particular Traffic Class (TC) is less than its minimum allocated bandwidth by allowing the difference to be available to other traffic classes.

After servicing the strict priority TCs, the amount of bandwidth (BW) left on the wire may be split among other TCs according to a minimal guarantee policy.

If, for instance, TC0 is set to 80% guarantee and TC1 to 20% (the TCs sum must be 100), then the BW left after servicing all strict priority TCs will be split according to this ratio.

Since this is a minimum guarantee, there is no maximum enforcement. This means, in the same example, that if TC1 did not use its share of 20%, the remainder will be used by TC0.

ETS is configured using the `mlnx_qos` tool ([mlnx_qos](#)) which allows you to:

- Assign a transmission algorithm to each TC (strict or ETS)
- Set minimal BW guarantee to ETS TCs

Usage:

```
mlnx_qos -i \[options\]
```

2.3.1.2.5.3 Rate Limit

Rate limit defines a maximum bandwidth allowed for a TC. Please note that 10% deviation from the requested values is considered acceptable.

2.3.1.2.5.4 Trust State

Trust state enables prioritizing sent/received packets based on packet fields.

The default trust state is PCP. Ethernet packets are prioritized based on the value of the field (PCP/DSCP).

For further information on how to configure Trust mode, please refer to [HowTo Configure Trust State on NVIDIA Adapters](#) community post.

Setting the Trust State mode shall be done before enabling SR-IOV in order to propagate the Trust State to the VFs.

2.3.1.2.5.5 Receive Buffer

By default, the receive buffer configuration is controlled automatically. Users can override the receive buffer size and receive buffer's xon and xoff thresholds using `mlnx_qos` tool. For further information, please refer to [HowTo Tune the Receive buffers on NVIDIA Adapters](#) community post.

2.3.1.2.5.6 DCBX Control Mode

DCBX settings, such as "ETS" and "strict priority" can be controlled by firmware or software. When DCBX is controlled by firmware, changes of QoS settings cannot be done by the software. The DCBX control mode is configured using the `mlnx_qos -d os/fw` command. For further information on how to configure the DCBX control mode, please refer to [mlnx_qos](#) community post.

2.3.1.2.6 Quality of Service Tools

2.3.1.2.6.1 mlnx_qos

`mlnx_qos` is a centralized tool used to configure QoS features of the local host. It communicates directly with the driver thus does not require setting up a DCBX daemon on the system.

The `mlnx_qos` tool enables the administrator of the system to:

- Inspect the current QoS mappings and configuration
The tool will also display maps configured by TC and `vconfig set_egress_map` tools, in order to give a centralized view of all QoS mappings.
- Set UP to TC mapping
- Assign a transmission algorithm to each TC (strict or ETS)
- Set minimal BW guarantee to ETS TCs
- Set rate limit to TCs
- Set DCBX control mode
- Set cable length
- Set trust state

For an unlimited ratelimit, set the ratelimit to 0.

Usage

```
mlnx_qos -i <interface> \[options\]
```

Options

<code>--version</code>	Show the program's version number and exit
<code>-h, --help</code>	Show this help message and exit

-f LIST, --pfc=LIST	Set priority flow control for each priority. LIST is a comma separated value for each priority starting from 0 to 7. Example: 0,0,0,0,1,1,1,1 enable PFC on TC4-7
-p LIST, --prio_tc=LIST	Maps UPs to TCs. LIST is 8 comma-separated TC numbers. Example: 0,0,0,0,1,1,1,1 maps UPs 0-3 to TC0, and UPs 4-7 to TC1
-s LIST, --tsa=LIST	Transmission algorithm for each TC. LIST is comma separated algorithm names for each TC. Possible algorithms: strict, ets and vendor. Example: vendor,strict,ets,ets,ets,ets,ets,ets sets TC0 to vendor, TC1 to strict, TC2-7 to ets
-t LIST, --tcbw=LIST	Set the minimally guaranteed %BW for ETS TCs. LIST is comma-separated percents for each TC. Values set to TCs that are not configured to ETS algorithm are ignored but must be present. Example: if TC0,TC2 are set to ETS, then 10,0,90,0,0,0,0,0 will set TC0 to 10% and TC2 to 90%. Percents must sum to 100
-r LIST, --ratelimit=LIST	Rate limit for TCs (in Gbps). LIST is a comma-separated Gbps limit for each TC. Example: 1,8,8 will limit TC0 to 1Gbps, and TC1,TC2 to 8 Gbps each
-d DCBX, --dcbx=DCBX	Set dcbx mode to firmware controlled(fw) or OS controlled(os). Note, when in OS mode, mlnx_qos should not be used in parallel with other dcbx tools, such as lldptool
--trust=TRUST	set priority trust state to pcp or dscp
--dscp2prio=DSCP2PRIO	Set/del a (dscp,prio) mapping. Example 'set,30,2' maps dscp 30 to priority 2. 'del,30,2' resets the dscp 30 mapping back to the default setting priority 0
--cable_len=CABLE_LEN	Set cable_len for buffer's xoff and xon thresholds
-i INTF, --interface=INTF	Interface name
-a	Show all interface's TCs

Get Current Configuration

```

ofed_scripts/utils/mlnx_qos -i ens1f0
DCBX mode: OS controlled
Priority trust state: dscp
dscp2prio mapping:
  prio:0 dscp:07,06,05,04,03,02,01,00,
  prio:1 dscp:15,14,13,12,11,10,09,08,
  prio:2 dscp:23,22,21,20,19,18,17,16,
  prio:3 dscp:31,30,29,28,27,26,25,24,
  prio:4 dscp:39,38,37,36,35,34,33,32,
  prio:5 dscp:47,46,45,44,43,42,41,40,
  prio:6 dscp:55,54,53,52,51,50,49,48,
  prio:7 dscp:63,62,61,60,59,58,57,56,
Cable len: 7
PFC configuration:
  priority 0 1 2 3 4 5 6 7
  enabled 0 0 0 0 0 0 0 0
tc: 0 ratelimit: unlimited, tsa: vendor
  priority: 1
tc: 1 ratelimit: unlimited, tsa: vendor
  priority: 0
tc: 2 ratelimit: unlimited, tsa: vendor
  priority: 2
tc: 3 ratelimit: unlimited, tsa: vendor
  priority: 3
tc: 4 ratelimit: unlimited, tsa: vendor
  priority: 4
tc: 5 ratelimit: unlimited, tsa: vendor
  priority: 5
tc: 6 ratelimit: unlimited, tsa: vendor
  priority: 6
tc: 7 ratelimit: unlimited, tsa: vendor
  priority: 7

```

Set ratelimit. 3Gbps for tc0 4Gbps for tc1 and 2Gbps for tc2

```
# mlx_qos -i <interface> -p 0,1,2 -r 3,4,2
tc: 0 ratelimit: 3 Gbps, tsa: strict
  up: 0
    skprio: 0
    skprio: 1
    skprio: 2 (tos: 8)
    skprio: 3
    skprio: 4 (tos: 24)
    skprio: 5
    skprio: 6 (tos: 16)
    skprio: 7
    skprio: 8
    skprio: 9
    skprio: 10
    skprio: 11
    skprio: 12
    skprio: 13
    skprio: 14
    skprio: 15
  up: 3
  up: 4
  up: 5
  up: 6
  up: 7
tc: 1 ratelimit: 4 Gbps, tsa: strict
  up: 1
tc: 2 ratelimit: 2 Gbps, tsa: strict
  up: 2
```

ConfigureQoS. Map UP0,7 to tc0,1,2,3 to tc1 and 4,5,6 to tc2. Set tc0,tc1 as ets and tc2 as strict. Divide ets 30% for tc0 and 70% for tc1

```
# mlx_qos -i <interface> -s ets,ets,strict -p 0,1,1,1,2,2,2 -t 30,70
tc: 0 ratelimit: 3 Gbps, tsa: ets, bw: 30%
  up: 0
    skprio: 0
    skprio: 1
    skprio: 2 (tos: 8)
    skprio: 3
    skprio: 4 (tos: 24)
    skprio: 5
    skprio: 6 (tos: 16)
    skprio: 7
    skprio: 8
    skprio: 9
    skprio: 10
    skprio: 11
    skprio: 12
    skprio: 13
    skprio: 14
    skprio: 15
  up: 7
tc: 1 ratelimit: 4 Gbps, tsa: ets, bw: 70%
  up: 1
  up: 2
  up: 3
tc: 2 ratelimit: 2 Gbps, tsa: strict
  up: 4
  up: 5
  up: 6
```

tc and tc_wrap.py

The tc tool is used to create 8 Traffic Classes (TCs).

The tool will either use the sysfs (/sys/class/net/<ethX>/qos/tc_num) or the tc tool to create the TCs.

Usage

```
tc_wrap.py -i <interface> \[options\]
```

Options

--version	show program's version number and exit
-h, --help	show this help message and exit
-u SKPRIO_UP, --skprio_up=SKPRIO_UP	maps sk_prio to priority for RoCE. LIST is <=16 comma separated priority. index of element is sk_prio
-i INTF, --interface=INTF	Interface name

Example

Run:

```
tc_wrap.py -i enp139s0
```

Output:

```
Tarrfic classes are set to 8
UP 0
  skprio: 0 (vlan 5)
UP 1
  skprio: 1 (vlan 5)
UP 2
  skprio: 2 (vlan 5 tos: 8)
UP 3
  skprio: 3 (vlan 5)
UP 4
  skprio: 4 (vlan 5 tos: 24)
UP 5
  skprio: 5 (vlan 5)
UP 6
  skprio: 6 (vlan 5 tos: 16)
UP 7
  skprio: 7 (vlan 5)
```

2.3.1.2.6.2 Additional Tools

tc tool compiled with the sch_mqprio module is required to support kernel v2.6.32 or higher. This is a part of iproute2 package v2.6.32-19 or higher. Otherwise, an alternative custom sysfs interface is available.

- mlnx_qos tool (package: ofed-scripts) requires python version 2.5 <= X
- tc_wrap.py (package: ofed-scripts) requires python version 2.5 <= X

2.3.1.3 Ethtool

Ethtool is a standard Linux utility for controlling network drivers and hardware, particularly for wired Ethernet devices. It can be used to:

- Get identification and diagnostic information
- Get extended device statistics
- Control speed, duplex, auto-negotiation and flow control for Ethernet devices
- Control checksum offload and other hardware offload features
- Control DMA ring sizes and interrupt moderation
- Flash device firmware using a .mfa2 image

Ethtool Supported Options

Options	Description
ethtool --set-priv-flags eth<x> <priv flag> <on/off>	Enables/disables driver feature matching the given private flag.
ethtool --show-priv-flags eth<x>	Shows driver private flags and their states (ON/OFF).
ethtool -a eth<x>	Queries the pause frame settings.
ethtool -A eth<x> [rx on off] [tx on off]	Sets the pause frame settings.
ethtool -c eth<x>	Queries interrupt coalescing settings.
ethtool -C eth<x> [pkt-rate- low N] [pkt-rate-high N] [rx-usecs-low N] [rx-usecs- high N]	Sets the values for packet rate limits and for moderation time high and low values.
ethtool -C eth<x> [rx-usecs N] [rx-frames N]	Sets the interrupt coalescing setting. rx-frames will be enforced immediately, rx-usecs will be enforced only when adaptive moderation is disabled. Note: usec settings correspond to the time to wait after the *last* packet is sent/received before triggering an interrupt.
ethtool -C eth<x> adaptive- rx on off	Enables/disables adaptive interrupt moderation. By default, the driver uses adaptive interrupt moderation for the receive path, which adjusts the moderation time to the traffic pattern.
ethtool -C eth<x> adaptive- tx on off	Note: Supported by mlx5e for ConnectX-4 and above adapter cards. Enables/disables adaptive interrupt moderation. By default, the driver uses adaptive interrupt moderation for the transmit path, which adjusts the moderation parameters (time/frames) to the traffic pattern.
ethtool -g eth<x>	Queries the ring size values.
ethtool -G eth<x> [rx <N>] [tx <N>]	Modifies the ring size.
ethtool -i eth<x>	Checks driver and device information. For example: <pre> driver: mlx5_core version: 5.1-0.4.0 firmware-version: 4.6.4046 (MT_QEMU000000) expansion-rom-version: bus-info: 0000:07:00.0 supports-statistics: yes supports-test: yes supports-eeprom-access: no supports-register-dump: no supports-priv-flags: yes </pre>
ethtool -k eth<x>	Queries the stateless offload status.

Options	Description
ethtool -K eth<x> [rx on off] [tx on off] [sg on off] [tso on off] [lro on off] [gro on off] [gso on off] [rxvlan on off] [txvlan on off] [ntuple on off] [rxhash on off] [rx-all on off] [rx-fcs on off]	Sets the stateless offload status. TCP Segmentation Offload (TSO), Generic Segmentation Offload (GSO): increase outbound throughput by reducing CPU overhead. It works by queuing up large buffers and letting the network interface card split them into separate packets. Large Receive Offload (LRO): increases inbound throughput of high-bandwidth network connections by reducing CPU overhead. It works by aggregating multiple incoming packets from a single stream into a larger buffer before they are passed higher up the networking stack, thus reducing the number of packets that have to be processed. LRO is available in kernel versions < 3.1 for untagged traffic. Hardware VLAN insertion Offload (txvlan): When enabled, the sent VLAN tag will be inserted into the packet by the hardware. Note: LRO will be done whenever possible. Otherwise GRO will be done. Generic Receive Offload (GRO) is available throughout all kernels. Hardware VLAN Striping Offload (rxvlan): When enabled received VLAN traffic will be stripped from the VLAN tag by the hardware. RX FCS (rx-fcs): Keeps FCS field in the received packets. Sets the stateless offload status. RX FCS validation (rx-all): Ignores FCS validation on the received packets.
ethtool -l eth<x>	Shows the number of channels.
ethtool -L eth<x> [rx <N>] [tx <N>]	Sets the number of channels. Notes: <ul style="list-style-type: none"> This also resets the RSS table to its default distribution, which is uniform across the cores on the NUMA (non-uniform memory access) node that is closer to the NIC. For ConnectX@-4 cards, use ethtool -L eth<x> combined <N> to set both RX and TX channels.
ethtool -m --dump-module-eprom eth<x> [raw on off] [hex on off] [offset N] [length N]	Queries/decodes the cable module eeprom information.
ethtool -p --identify DEVNAME	Enables visual identification of the port by LED blinking [TIME-IN-SECONDS].
ethtool -p --identify eth<x> <LED duration>	Allows users to identify interface's physical port by turning the ports LED on for a number of seconds. Note: The limit for the LED duration is 65535 seconds.
ethtool -S eth<x>	Obtains additional device statistics.

Options	Description																						
ethtool -s eth<x> advertise <N> autoneg on	<p>Changes the advertised link modes to requested link modes <N>. To check the link modes' hex values, run <code><man ethtool></code> and to check the supported link modes, run <code>ethtool eth<x></code>.</p> <p>For advertising new link modes, make sure to configure the entire bitmap as follows:</p> <table border="1"> <tbody> <tr> <td>200GAUI-4 / 200GBASE-CR4/KR4</td> <td>0x7c00000000000000</td> </tr> <tr> <td>100GAUI-2 / 100GBASE-CR2 / KR2</td> <td>0x3E00000000000000</td> </tr> <tr> <td>CAUI-4 / 100GBASE-CR4 / KR4</td> <td>0xF000000000</td> </tr> <tr> <td>50GAUI-1 / LAUI-1 / 50GBASE-CR / KR</td> <td>0x1F00000000000000</td> </tr> <tr> <td>50GAUI-2 / LAUI-2 / 50GBASE-CR2/KR2</td> <td>0x10C000000000</td> </tr> <tr> <td>XLAUI-4/XLPPI-4 // 40G</td> <td>0x78000000</td> </tr> <tr> <td>25GAUI-1 / 25GBASE-CR / KR</td> <td>0x3800000000</td> </tr> <tr> <td>XFI / XAUI-1 // 10G</td> <td>0x7C0000181000</td> </tr> <tr> <td>5GBASE-R</td> <td>0x10000000000000</td> </tr> <tr> <td>2.5GBASE-X / 2.5GMII</td> <td>0x820000000000</td> </tr> <tr> <td>1000BASE-X / SGMII</td> <td>0x20000020020</td> </tr> </tbody> </table> <p>Notes:</p> <ul style="list-style-type: none"> Both previous and new link modes configurations are supported, however, they must be run separately. Any link mode configuration on Kernels below v5.1 and ConnectX-6 HCAs will result in the advertisement of the full capabilities. <code><autoneg on></code> only sends a hint to the driver that the user wants to modify advertised link modes and not speed. 	200GAUI-4 / 200GBASE-CR4/KR4	0x7c00000000000000	100GAUI-2 / 100GBASE-CR2 / KR2	0x3E00000000000000	CAUI-4 / 100GBASE-CR4 / KR4	0xF000000000	50GAUI-1 / LAUI-1 / 50GBASE-CR / KR	0x1F00000000000000	50GAUI-2 / LAUI-2 / 50GBASE-CR2/KR2	0x10C000000000	XLAUI-4/XLPPI-4 // 40G	0x78000000	25GAUI-1 / 25GBASE-CR / KR	0x3800000000	XFI / XAUI-1 // 10G	0x7C0000181000	5GBASE-R	0x10000000000000	2.5GBASE-X / 2.5GMII	0x820000000000	1000BASE-X / SGMII	0x20000020020
200GAUI-4 / 200GBASE-CR4/KR4	0x7c00000000000000																						
100GAUI-2 / 100GBASE-CR2 / KR2	0x3E00000000000000																						
CAUI-4 / 100GBASE-CR4 / KR4	0xF000000000																						
50GAUI-1 / LAUI-1 / 50GBASE-CR / KR	0x1F00000000000000																						
50GAUI-2 / LAUI-2 / 50GBASE-CR2/KR2	0x10C000000000																						
XLAUI-4/XLPPI-4 // 40G	0x78000000																						
25GAUI-1 / 25GBASE-CR / KR	0x3800000000																						
XFI / XAUI-1 // 10G	0x7C0000181000																						
5GBASE-R	0x10000000000000																						
2.5GBASE-X / 2.5GMII	0x820000000000																						
1000BASE-X / SGMII	0x20000020020																						
ethtool -s eth<x> msglvl [N]	Changes the current driver message level.																						
ethtool -s eth<x> speed <SPEED> autoneg off	<p>Changes the link speed to requested <SPEED>. To check the supported speeds, run <code>ethtool eth<x></code>.</p> <p>Note: <code><autoneg off></code> does not set autoneg OFF, it only hints the driver to set a specific speed.</p>																						
ethtool -t eth<x>	Performs a self-diagnostics test.																						
ethtool -T eth<x>	Shows time stamping capabilities																						
ethtool -x eth<x>	Retrieves the receive flow hash indirection table.																						
ethtool -X eth<x> equal a b c...	<p>Sets the receive flow hash indirection table.</p> <p>Note: The RSS table configuration is reset whenever the number of channels is modified (using <code>ethtool -L</code> command).</p>																						
ethtool --show-fec eth<x>	<p>Queries current Forward Error Correction (FEC) encoding in case FEC is supported.</p> <p>Note: An output of "baser" implies Firecode encoding.</p>																						
ethtool --set-fec eth<x> encoding auto off rs baser	<p>Configures Forward Error Correction (FEC).</p> <p>Note: 'baser' encoding applies to the Firecode encoding, and 'auto' regards the HCA's default.</p>																						
ethtool -f --flash <devname> FILE [N]	Flash firmware image on the device using the specified .mfa2 file (FILE). By default, the command flashes all the regions on the device unless a region number (N) is specified.																						

2.3.1.4 Checksum Offload

The following Receive IP/L4 Checksum Offload modes are supported.

- **CHECKSUM_UNNECESSARY**: When this mode is used, the driver indicates to the Linux Networking Stack that the hardware successfully validated the IP and L4 checksum so the Linux Networking Stack does not need to deal with IP/L4 Checksum validation.
- **CHECKSUM_COMPLETE**: When this mode is used, the driver still reports to the OS the calculated by hardware checksum value. This allows accelerating checksum validation in Linux Networking Stack, since it does not have to calculate the whole checksum including payload by itself.
- **CHECKSUM_NONE**: When this mode is used, the driver indicates to the Linux Networking Stack that it must calculate and validate the IP/L4 checksum.

2.3.1.5 Ignore Frame Check Sequence (FCS) Errors

Upon receiving packets, the packets go through a checksum validation process for the FCS field. If the validation fails, the received packets are dropped.

When FCS is enabled (disabled by default), the device does not validate the FCS field even if the field is invalid.

It is not recommended to enable FCS.

For further information on how to enable/disable FCS, please refer to [ethtool option rx-fcs on/off](#).

2.3.1.6 RDMA over Converged Ethernet (RoCE)

Remote Direct Memory Access (RDMA) is the remote memory management capability that allows server-to-server data movement directly between application memory without any CPU involvement. RDMA over Converged Ethernet (RoCE) is a mechanism to provide this efficient data transfer with very low latencies on lossless Ethernet networks. With advances in data center convergence over reliable Ethernet, ConnectX® Ethernet adapter cards family with RoCE uses the proven and efficient RDMA transport to provide the platform for deploying RDMA technology in mainstream data center application at 10GigE and 40GigE link-speed. ConnectX® Ethernet adapter cards family with its hardware offload support takes advantage of this efficient RDMA transport (InfiniBand) services over Ethernet to deliver ultra-low latency for performance-critical and transaction-intensive applications such as financial, database, storage, and content delivery networks.

When working with RDMA applications over Ethernet link layer the following points should be noted:

- The presence of a Subnet Manager (SM) is not required in the fabric. Thus, operations that require communication with the SM are managed in a different way in RoCE. This does not affect the API but only the actions such as joining the multicast group, that need to be taken when using the API
- Since LID is a layer 2 attribute of the InfiniBand protocol stack, it is not set for a port and is displayed as zero when querying the port
- With RoCE, the alternate path is not set for RC QP. Therefore, APM (another type of High Availability and part of the InfiniBand protocol) is not supported

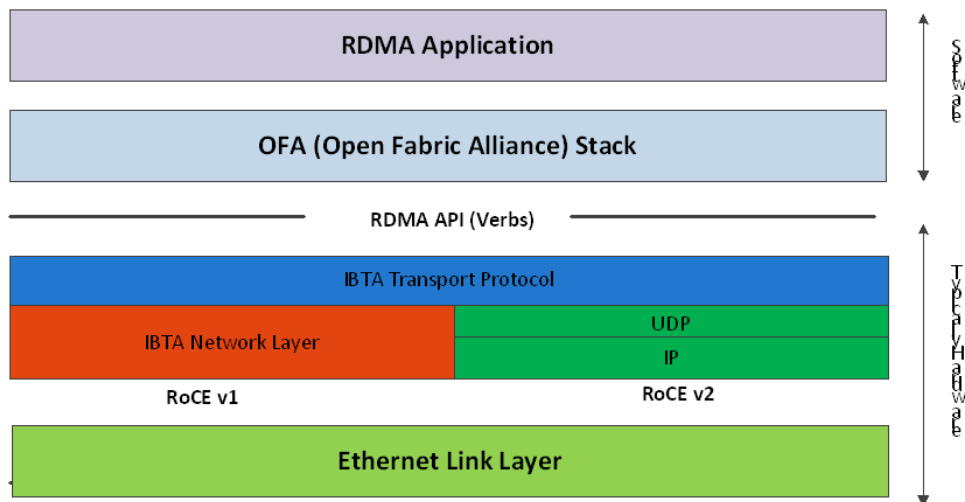
- Since the SM is not present, querying a path is impossible. Therefore, the path record structure must be filled with relevant values before establishing a connection. Hence, it is recommended working with RDMA-CM to establish a connection as it takes care of filling the path record structure
- VLAN tagged Ethernet frames carry a 3-bit priority field. The value of this field is derived from the IB SL field by taking the 3 least significant bits of the SL field
- RoCE traffic is not shown in the associated Ethernet device's counters since it is offloaded by the hardware and does not go through Ethernet network driver. RoCE traffic is counted in the same place where InfiniBand traffic is counted; /sys/class/infiniband/<device>/ports/<port number>/counters/

2.3.1.6.1 RoCE Modes

RoCE encapsulates IB transport in one of the following Ethernet packets:

- RoCEv1 - dedicated ether type (0x8915)
- RoCEv2 - UDP and dedicated UDP port (4791)

RoCEv1 and RoCEv2 Protocol Stack



2.3.1.6.1.1 RoCEv1

RoCE v1 protocol is defined as RDMA over Ethernet header (as shown in the figure above). It uses ethertype 0x8915 and can be used with or without the VLAN tag. The regular Ethernet MTU applies on the RoCE frame.

2.3.1.6.1.2 RoCEv2

A straightforward extension of the RoCE protocol enables traffic to operate in IP layer 3 environments. This capability is obtained via a simple modification of the RoCE packet format. Instead of the GRH used in RoCE, IP routable RoCE packets carry an IP header which allows traversal of IP L3 Routers and a UDP header (RoCEv2 only) that serves as a stateless encapsulation layer for the RDMA Transport Protocol Packets over IP.

The proposed RoCEv2 packets use a well-known UDP destination port value that unequivocally distinguishes the datagram. Similar to other protocols that use UDP encapsulation, the UDP source port field is used to carry an opaque flow-identifier that allows network devices to implement packet forwarding optimizations (e.g. ECMP) while staying agnostic to the specifics of the protocol header format.

Furthermore, since this change exclusively affects the packet format on the wire, and due to the fact that with RDMA semantics packets are generated and consumed below the AP, applications can seamlessly operate over any form of RDMA service, in a completely transparent way.

Both RoCEv1 and RoCEv2 are supported by default; the driver associates all GID indexes to RoCEv1 and RoCEv2, thus, a single entry for each RoCE version.

For further information, please refer to [Recommended Network Configuration Examples For RoCE Deployment](#) community post.

2.3.1.6.2 GID Table Population

GID table entries are created whenever an IP address is configured on one of the Ethernet devices of the NIC's ports. Each entry in the GID table for RoCE ports has the following fields:

- GID value
- GID type
- Network device

The GID table is occupied with two GIDs, both with the same GID value but with different types. The network device in an entry is the Ethernet device with the IP address that GID is associated with. The GID format can be of 2 types; IPv4 and IPv6. IPv4 GID is an IPv4-mapped IPv6 address, while IPv6 GID is the IPv6 address itself. Layer 3 header for packets associated with IPv4 GIDs will be IPv4 (for RoCEv2) and IPv6/GRH for packets associated with IPv6 GIDs and IPv4 GIDs for RoCEv1.

GID Table in sysfs

GID table is exposed to userspace via sysfs

- GID values can be read from:

```
/sys/class/infiniband/{device}/ports/{port}/gids/{index}
```

- GID type can be read from:

```
/sys/class/infiniband/{device}/ports/{port}/gid_attrs/types/{index}
```

- GID net_device can be read from:

```
/sys/class/infiniband/{device}/ports/{port}/gid_attrs/ndevs/{index}
```

2.3.1.6.2.1 Setting the RoCE Mode for a Queue Pair (QP)

Setting RoCE mode for devices that support two RoCE modes is different for RC/UC QPs (connected QP types) and UD QP.

To modify an RC/UC QP (connected QP) from INIT to RTR, an Address Vector (AV) must be given. The AV, among other attributes, should specify the index of the port's GID table for the source GID of the QP. The GID type in that index will be used to set the RoCE type of the QP.

2.3.1.6.2.2 Setting RoCE Mode of RDMA_CM Applications

RDMA_CM interface requires only the active side of the peer to pass the IP address of the passive side. The RDMA_CM decides upon the source GID to be used and obtains it from the GID table. Since more than one instance of the GID value is possible, the lookup should be also according to the GID type. The type to use for the lookup is defined as a global value of the RDMA_CM module. Changing the value of the GID type for the GID table lookups is done using the `cma_roce_mode` script.

To print the current RoCE mode for a device port:

```
cma_roce_mode -d <dev> -p <port>
```

To set the RoCE mode for a device port:

```
cma_roce_mode -d <dev> -p <port> -m <1|2>
```

2.3.1.6.2.3 GID Table Example

The following is an example of the GID table.

DEV	PORT	INDEX	GID	IPv4	Type	Netdev
mlx5_0	1	0	fe80:0000:0000:0000:ba59:9fff:fe1a:e3ea		v1	p4p1
mlx5_0	1	1	fe80:0000:0000:0000:ba59:9fff:fe1a:e3ea		v2	p4p1
mlx5_0	1	2	0000:0000:0000:0000:0000:ffff:0a0a:0a01	10.10.10.1	v1	p4p1
mlx5_0	1	3	0000:0000:0000:0000:0000:ffff:0a0a:0a01	10.10.10.1	v2	p4p1
mlx5_1	1	0	fe80:0000:0000:0000:ba59:9fff:fe1a:e3eb		v1	p4p2
mlx5_1	1	1	fe80:0000:0000:0000:ba59:9fff:fe1a:e3eb		v2	p4p2

where:

- Entries on port 1 index 0/1 are the default GIDs, one for each supported RoCE type
- Entries on port 1 index 2/3 belong to IP address 192.168.1.70 on eth1
- Entries on port 1 index 4/5 belong to IP address 193.168.1.70 on eth1.100

- Packets from a QP that is associated with these GID indexes will have a VLAN header (VID=100)
- Entries on port 1 index 6/7 are IPv6 GID. Packets from a QP that is associated with these GID indexes will have an IPv6 header

2.3.1.6.3 RoCE Lossless Ethernet Configuration

In order to function reliably, RoCE requires a form of flow control. While it is possible to use global flow control, this is normally undesirable, for performance reasons.

The normal and optimal way to use RoCE is to use Priority Flow Control (PFC). To use PFC, it must be enabled on all endpoints and switches in the flow path.

2.3.1.6.3.1 Configuring SwitchX® Based Switch System

To enable RoCE, the SwitchX should be configured as follows:

- Ports facing the host should be configured as access ports, and either use global pause or Port Control Protocol (PCP) for priority flow control
- Ports facing the network should be configured as trunk ports, and use Port Control Protocol (PCP) for priority flow control
- For further information on how to configure SwitchX, please refer to SwitchX User Manual

2.3.1.6.4 Installing and Loading the Driver

To install and load the driver:

1. Install MLNX_OFED (See [Installation](#) section for further details).

RoCE is installed as part of mlx5 and other modules upon driver's installation.

The list of the modules that will be loaded automatically upon boot can be found in the configuration file `/etc/infiniband/openib.conf`.

2. Query for the device's information. Example:

```
ibv_devinfo MLNX_OFED_LINUX-5.0-2.1.8.0:
```

3. Display the existing MLNX_OFED version.

```
ofed_info -s
hca_id: mlx5_0
  transport: InfiniBand (0)
  fw_ver: 16.28.0578
  node_guid: ec0d:9a03:0044:3764
  sys_image_guid: ec0d:9a03:0044:3764
  vendor_id: 0x02c9
  vendor_part_id: 4121
  hw_ver: 0x0
  board_id: MT_000000009
  phys_port_cnt: 1
    port: 1
      state: PORT_ACTIVE (4)
      max_mtu: 4096 (5)
      active_mtu: 1024 (3)
      sm_lid: 0
      port_lid: 0
      port_lmc: 0x00
      link_layer: Ethernet
```

Output Notes:

The port's state is: Ethernet is in PORT_ACTIVE state	The port state can also be obtained by running the following command: <code># cat /sys/class/infiniband/mlx5_0/ports/1/state 4: ACTIVE</code>
link_layer parameter shows that port 1 is Ethernet	The link_layer can also be obtained by running the following command: <code># cat /sys/class/infiniband/mlx5_0/ports/1/link_layer Ethernet</code>
The fw_ver parameter shows that the firmware version is 16.28.0578.	The firmware version can also be obtained by running the following command: <code># cat /sys/class/infiniband/mlx5_0/fw_ver 16.28.0578</code>

2.3.1.6.4.1 Associating InfiniBand Ports to Ethernet Ports

The `mlx5_ib` driver holds a reference to the net device for getting notifications about the state of the port, as well as using the `mlx5_core` driver to resolve IP addresses to MAC that are required for address vector creation. However, RoCE traffic does not go through the `mlx5_core` driver; it is completely offloaded by the hardware.

```
# ibdev2netdev
mlx5_0 port 1 <====> eth2
#
```

2.3.1.6.4.2 Configuring an IP Address to the netdev Interface

To configure an IP address to the netdev interface:

1. Configure an IP address to the netdev interface on both sides of the link.

```
# ifconfig eth2 20.4.3.220
# ifconfig eth2
eth2      Link encap:Ethernet HWaddr 00:02:C9:08:E8:11
          inet addr:20.4.3.220 Bcast:20.255.255.255 Mask:255.0.0.0
          UP BROADCAST MULTICAST MTU:1500 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b) TX bytes:0 (0.0 b)
```

2. Make sure that ping is working.

```
ping 20.4.3.219
PING 20.4.3.219 (20.4.3.219) 56(84) bytes of data.
64 bytes from 20.4.3.219: icmp_seq=1 ttl=64 time=0.873 ms
64 bytes from 20.4.3.219: icmp_seq=2 ttl=64 time=0.198 ms
64 bytes from 20.4.3.219: icmp_seq=3 ttl=64 time=0.167 ms
20.4.3.219 ping statistics -
3 packets transmitted, 3 received, 0% packet loss, time 2000ms rtt min/avg/max/mdev = 0.167/0.412/0.873/0.326 ms
```

2.3.1.6.4.3 Adding VLANs

To add VLANs:

1. Make sure that the 8021q module is loaded.

```
modprobe 8021q
```

2. Add VLAN.

```
# vconfig add eth2 7
Added VLAN with VID == 7 to IF -:eth2:-
#
```

3. Configure an IP address.

```
ifconfig eth2.7 7.4.3.220
```

2.3.1.6.4.4 Defining Ethernet Priority (PCP in 802.1q Headers)

1. Define Ethernet priority on the server.

```
# ibv_rc_pingpong -g 1 -i 2 -l 4
local address: LID 0x0000, QPN 0x1c004f, PSN 0x9daf6c, GID fe80::202:c900:708:e799
remote address: LID 0x0000, QPN 0x1c004f, PSN 0xb0a49b, GID fe80::202:c900:708:e811
8192000 bytes in 0.01 seconds = 4840.89 Mbit/sec
1000 iters in 0.01 seconds = 13.54 usec/iter
```

2. Define Ethernet priority on the client.

```
# ibv_rc_pingpong -g 1 -i 2 -l 4 sw419
local address: LID 0x0000, QPN 0x1c004f, PSN 0xb0a49b, GID fe80::202:c900:708:e811
remote address: LID 0x0000, QPN 0x1c004f, PSN 0x9daf6c, GID fe80::202:c900:708:e799
8192000 bytes in 0.01 seconds = 4855.96 Mbit/sec
1000 iters in 0.01 seconds = 13.50 usec/iter
```

2.3.1.6.4.5 Using rdma_cm Tests

1. Use rdma_cm test on the server.

```
# ucmatose
cmatose: starting server
initiating data transfers
completing sends
receiving data transfers
data transfers complete
cmatose: disconnecting
disconnected
test complete
return status 0
#
```

2. Use rdma_cm test on the client.

```
# ucmatose -s 20.4.3.219
cmatose: starting client
cmatose: connecting
receiving data transfers
sending replies
data transfers complete
test complete
return status 0
#
```

This server-client run is without PCP or VLAN because the IP address used does not belong to a VLAN interface. If you specify a VLAN IP address, then the traffic should go over VLAN.

2.3.1.6.5 Type Of Service (ToS)

2.3.1.6.5.1 Overview

The TOS field for rdma_cm sockets can be set using the rdma_set_option() API, just as it is set for regular sockets. If a TOS is not set, the default value (0) is used. Within the rdma_cm kernel driver, the TOS field is converted into an SL field. The conversion formula is as follows:

- $SL = TOS \gg 5$ (e.g., take the 3 most significant bits of the TOS field)

In the hardware driver, the SL field is converted into PCP by the following formula:

- $PCP = SL \& 7$ (take the 3 least significant bits of the TOS field)

SL affects the PCP only when the traffic goes over tagged VLAN frames.

2.3.1.6.5.2 DSCP

A new entry has been added to the RDMA-CM configs that allows users to select default TOS for RDMA-CM QPs. This is useful for users that want to control the TOS field without changing their code. Other applications that set the TOS explicitly using the rdma_set_option API will continue to work as expected to override the configs value.

For further information about DSCP marking, refer to [HowTo Set Egress ToS/DSCP on RDMA- CM QPs](#) Community post.

2.3.1.6.6 RoCE LAG

RoCE LAG is a feature meant for mimicking Ethernet bonding for IB devices and is available for dual port cards only.

This feature is supported on kernel versions 4.9 and above.

RoCE LAG mode is entered when both Ethernet interfaces are configured as a bond in one of the following modes:

- active-backup (mode 1)
- balance-xor (mode 2)
- 802.3ad (LACP) (mode 4)

Any change of bonding configuration that negates one of the above rules (i.e, bonding mode is not 1, 2 or 4, or both Ethernet interfaces that belong to the same card are not the only slaves of the bond interface), will result in exiting RoCE LAG mode and the return to normal IB device per port configuration.

Once RoCE LAG is enabled, instead of having two IB devices; mlx5_0 and mlx5_1, there will be one device named mlx5_bond_0.

For information on how to configure RoCE LAG, refer to [HowTo Configure RoCE over LAG \(ConnectX-4/ConnectX-5/ConnectX-6\)](#) Community post.

2.3.1.6.7 Disabling RoCE

By default, RoCE is enabled on all mlx5 devices. When RoCE is enabled, all traffic to UDP port 4791 is treated as RoCE traffic by the device.

In case you are only interested in Ethernet (no RDMA) and wish to enable forwarding of traffic to this port, you can disable RoCE through sysfs:

```
echo <0|1> > /sys/devices/{pci-bus-address}/roce_enable
```

Once RoCE is disabled, only Ethernet traffic will be supported. Therefore, there will be no GID tables and only Raw Ethernet QPs will be supported.

The current RoCE state can be queried by sysfs:

```
cat /sys/devices/{pci-bus-address}/roce_enable
```

2.3.1.6.8 Enabling/Disabling RoCE on VMs via VFs

By default, when configuring VFs on the hypervisor, all VFs will be enabled with RoCE. This means they require more OS memory (from the VM). In case you are only interested in Ethernet (no RDMA) on the VM, and you wish to save the VM memory, you can disable RoCE on the VF from the hypervisor. In addition, by disabling RoCE, a VM can have the capability of utilizing the RoCE UDP port (4791) for standard UDP traffic.

For details on how to enable/disable RoCE on a VF, refer to [HowTo Enable/Disable RoCE on VMs via VFs](#) Community post.

2.3.1.6.9 Force DSCP

This feature enables setting a global traffic_class value for all RC QPs, or setting a specific traffic class based on several matching criteria.

Usage

- To set a single global traffic class to be applied to all QPs, write the desired global traffic_class value to /sys/class/infiniband/<dev>/tc/<port>/traffic_class.

Note the following:

- Negative values indicate that the feature is disabled. traffic_class value can be set using `ibv_modify_qp()`
- Valid values range between 0 - 255

The ToS field is 8 bits, while the DSCP field is 6 bits. To set a DSCP value of X, you need to multiply this value by 4 (SHIFT 2). For example, to set DSCP value of 24, set the ToS bit to 96 (24x4=96).

- To set multiple traffic class values based on source and/or destination IPs, write the desired rule to `/sys/class/infiniband/<dev>/tc/<port>/traffic_class`. For example:

```
echo "tclass=16,src_ip=1.1.1.2,dst_ip=1.1.1.0/24" > /sys/class/infiniband/mlx5_0/tc/1/traffic_class
```

Note: Adding "tclass" prefix to tclass value is optional.

In the example above, traffic class 16 will be set to any QP with source IP 1.1.1.2 and destination IP 1.1.1.0/24.

Note that when setting a specific traffic class, the following rule precedence will apply:

- If a global traffic class value is set, it will be applied to all QPs
- If no global traffic class value is set, and there is a rule with matching source and destination IPs applicable to at least one QP, it will be applied
- Rules only with matching source and/or destination IPs have no defined precedence over other rules with matching source and/or destination IPs

Notes:

- A mask can be provided when using destination IPv4 addresses
- The rule precedence is not affected by the order in which rules are inserted
- Overlapping rules are entirely up to the administrator.
- "tclass=-1" will remove the rule from the database

2.3.1.6.10 Force Time to Live (TTL)

This feature enables setting a global TTL value for all RC QPs.

Write the desired TTL value to `/sys/class/infiniband/<dev>/tc/<port>/ttl`. Valid values range between 0 - 255

2.3.1.6.11 Out-of-Order (OOO)

In certain fabric configurations, RoCE traffic packets for a given QP may take up different paths in a network from the source to the destination. This results in packets being received in an out-of-order manner. In order to avoid retransmission, these packets can now be handled instead of being dropped by the doing the following:

- Achieving better network utilization
- Decreasing latency

Data will be placed into host memory in an out-of-order (OOO) manner when out-of-order messages are received.

For information on how to set up out-of-order processing over RoCE, contact support.

2.3.1.7 Flow Control

2.3.1.7.1 Priority Flow Control (PFC)

Priority Flow Control (PFC) IEEE 802.1Qbb applies pause functionality to specific classes of traffic on the Ethernet link. For example, PFC can provide lossless service for the RoCE traffic and best-effort

service for the standard Ethernet traffic. PFC can provide different levels of service to specific classes of Ethernet traffic (using IEEE 802.1p traffic classes).

2.3.1.7.1.1 Configuring PFC on ConnectX-4 and above

1. Enable PFC on the desired priority:

```
mlnx_qos -i <ethX> --pfc <0/1>,<0/1>,<0/1>,<0/1>,<0/1>,<0/1>,<0/1>,<0/1>
```

Example (Priority=4):

```
mlnx_qos -i eth1 --pfc 0,0,0,0,1,0,0,0
```

2. Create a VLAN interface:

```
vconfig add <ethX> <VLAN_ID>
```

Example (VLAN_ID=5):

```
vconfig add eth1 5
```

3. Set egress mapping:

- a. For Ethernet traffic:

```
vconfig set_egress_map <vlan_einterface> <skprio> <up>
```

Example (skprio=3, up=5):

```
vconfig set_egress_map eth1.5 3 5
```

4. Create 8 Traffic Classes (TCs):

```
tc_wrap.py -i <interface>
```

5. Enable PFC on the switch.

For information on how to enable PFC on your respective switch, please refer to Switch FC/ PFC Configuration sections in the [RDMA/RoCE Solutions](#) Community page.

2.3.1.7.1.2 PFC Configuration Using LLDP DCBX

PFC Configuration on Hosts

PFC Auto-Configuration Using LLDP Tool in the OS

1. Start lldpad daemon on host.

```
lldpad -d Or  
service lldpad start
```

2. Send lldpad packets to the switch.

```
lldptool set-lldp -i <ethX> adminStatus=rxtx ;  
lldptool -T -i <ethX> -V sysName enableTx=yes ;  
lldptool -T -i <ethX> -V portDesc enableTx=yes ;  
lldptool -T -i <ethX> -V sysDesc enableTx=yes  
lldptool -T -i <ethX> -V sysCap enableTx=yess
```

```
lldptool -T -i <ethX> -V mngAddr enableTx=yes
lldptool -T -i <ethX> -V PFC enableTx=yes;
lldptool -T -I <ethX> -V CEE-DCBX enableTx=yes;
```

3. Set the PFC parameters.

- For the CEE protocol, use dcbtool:

```
dcbtool sc <ethX> pfc pfcup:<xxxxxxxx>
```

Example:

```
dcbtool sc eth6 pfc pfcup:01110001
```

where:

[pfcup:xx xxxxxx]	Enables/disables priority flow control. From left to right (priorities 0-7) - x can be equal to either 0 or 1. 1 indicates that the priority is configured to transmit priority pause.
----------------------	--

- For IEEE protocol, use lldptool:

```
lldptool -T -i <ethX> -V PFC enabled=x,x,x,x,x,x,x,x
```

Example:

```
lldptool -T -i eth2 -V PFC enabled=1,2,4
```

where:

enabled	Displays or sets the priorities with PFC enabled. The set attribute takes a comma-separated list of priorities to enable, or the string none to disable all priorities.
---------	---

PFC Auto-Configuration Using LLDP in the Firmware (for mlx5 driver)

There are two ways to configure PFC and ETS on the server:

- Local Configuration - Configuring each server manually.
- Remote Configuration - Configuring PFC and ETS on the switch, after which the switch will pass the configuration to the server using LLDP DCBX TLVs.

There are two ways to implement the remote configuration using mlx5 driver:

- Configuring the adapter firmware to enable DCBX.
- Configuring the host to enable DCBX.

For further information on how to auto-configure PFC using LLDP in the firmware, refer to the [HowTo Auto-Config PFC and ETS on ConnectX-4 via LLDP DCBX](#) Community post.

PFC Configuration on Switches

- In order to enable DCBX, LLDP should first be enabled:

```
switch (config) # lldp
show lldp interfaces ethernet remote
```

- Add DCBX to the list of supported TLVs per required interface.

For IEEE DCBX:

```
switch (config) # interface 1/1
switch (config interface ethernet 1/1) # lldp tlv-select dcbx
```

For CEE DCBX:

```
switch (config) # interface 1/1
switch (config interface ethernet 1/1) # lldp tlv-select dcbx-cee
```

3. [Optional] Application Priority can be configured on the switch, with the required ethertype and priority. For example, IP packet, priority 1:

```
switch (config) # dcb application-priority 0x8100 1
```

4. Make sure PFC is enabled on the host (for enabling PFC on the host, refer to [PFC Configuration on Hosts](#) section above). Once it is enabled, it will be passed in the LLDP TLVs.
5. Enable PFC with the desired priority on the Ethernet port.

```
dcb priority-flow-control enable force
dcb priority-flow-control priority <priority> enable
interface ethernet <port> dcb priority-flow-control mode on force
```

Example - Enabling PFC with priority 3 on port 1/1:

```
dcb priority-flow-control enable force
dcb priority-flow-control priority 3 enable
interface ethernet 1/1 dcb priority-flow-control mode on force
```

Priority Counters

Several ingress and egress counters per priority are supported. Run `ethtool -S` to get the full list of port counters.

ConnectX-4 Counters

- Rx and Tx Counters:
 - Packets
 - Bytes
 - Octets
 - Frames
 - Pause
 - Pause frames
 - Pause Duration
 - Pause Transition

ConnectX-4 Example

```
# ethtool -S eth35 | grep prio4
prio4_rx_octets: 62147780800
prio4_rx_frames: 14885696
prio4_tx_octets: 0
prio4_tx_frames: 0
prio4_rx_pause: 0
prio4_rx_pause_duration: 0
prio4_tx_pause: 26832
prio4_tx_pause_duration: 14508
prio4_rx_pause_transition: 0
```

Note: The Pause counters in ConnectX-4 are visible via `ethtool` only for priorities on which PFC is enabled.

2.3.1.7.1.3 PFC Storm Prevention

PFC storm prevention enables toggling between default and auto modes.

The stall prevention timeout is configured to 8 seconds by default. Auto mode sets the stall prevention timeout to be 100 msec.

The feature can be controlled using sysfs in the following directory: `/sys/class/net/eth*/settings/pfc_stall_prevention`

- To query the PFC stall prevention mode:

```
cat /sys/class/net/eth*/settings/pfc_stall_prevention
```

Example

```
$ cat /sys/class/net/ens6/settings/pfc_stall_prevention
default
```

- To configure the PFC stall prevention mode:

```
Echo "auto"/"default" > /sys/class/net/eth*/settings/pfc_stall_prevention
```

The following two counters were added to the ethtool -S:

- `tx_Pause_storm_warning_events` - when the device is stalled for a period longer than a pre-configured watermark, the counter increases, allowing the debug utility an insight into current device status.
- `tx_pause_storm_error_events` - when the device is stalled for a period longer than a pre-configured timeout, the pause transmission is disabled, and the counter increase.

2.3.1.7.2 Dropless Receive Queue (RQ)

Dropless RQ feature enables the driver to notify the FW when SW receive queues are overloaded.

This scenario takes place when the handling of SW receive queue is slower than the handling of the HW receive queues.

When this feature is enabled, a packet that is received while the receive queue is full will not be immediately dropped. The FW will accumulate these packets assuming posting of new WQEs will resume shortly. If received WQEs are not posted after a certain period of time, `out_of_buffer` counter will increase, indicating that the packet has been dropped.

This feature is disabled by default. In order to activate it, ensure that Flow Control feature is also enabled.

➤ *To enable the feature, run:*

```
ethtool --set-priv-flags ens6 dropless_rq on
```

➤ *To get the feature state, run:*


```
ethtool --show-priv-flags DEVNAME
```

Output example:

```
Private flags for DEVNAME:
rx_cqe_moder      : on
rx_cqe_compress  : off
sniffer           : off
dropless_rq      : off
hw_lro            : off
```

➤ *To disable the feature, run:*

```
ethtool --set-priv-flags ens6 dropless_rq off
```

2.3.1.8 Explicit Congestion Notification (ECN)

ECN is an extension to the IP protocol. It allows reliable communication by notifying all ends of communication when congestion occurs. This is done without dropping packets.

Please note that this feature requires all nodes in the path (nodes, routers etc) between the communicating nodes to support ECN to ensure reliable communication. ECN is marked as 2 bits in the traffic control IP header. This ECN implementation refers to RoCE v2.

2.3.1.8.1 Enabling ECN

➤ *To enable ECN on the hosts:*

1. Enable ECN in sysfs.

```
/sys/class/net/<interface>/<protocol>/ecn-<protocol>_enable =1
```

2. Query the attribute.

```
cat /sys/class/net/<interface>/ecn/<protocol>/params/<requested attribute>
```

3. Modify the attribute.

```
echo <value> /sys/class/net/<interface>/ecn/<protocol>/params/<requested attribute>
```

ECN supports the following algorithms:

- r_roce_ecn_rp - Reaction point
- r_roce_ecn_np - Notification point

Each algorithm has a set of relevant parameters and statistics, which are defined per device, per port, per priority.

➤ *To query whether ECN is enabled per Priority X:*

```
cat /sys/class/net/<interface>/ecn/<protocol>/enable/X
```

➤ *To read ECN configurable parameters:*

```
cat /sys/class/net/<interface>/ecn/<protocol>/requested attributes
```

➤ *To enable ECN for each priority per protocol:*

```
echo 1 > /sys/class/net/<interface>/ecn/<protocol>/enable/X
```

➤ *To modify ECN configurable parameters:*

```
echo <value> > /sys/class/net/<interface>/ecn/<protocol>/requested attributes
```

where:

- X: priority {0..7}
- protocol: roce_rp / roce_np
- requested attributes: Next Slide for each protocol.

2.3.1.9 RSS Support

2.3.1.9.1 RSS Hash Function

The device has the ability to use XOR as the RSS distribution function, instead of the default Toeplitz function.

The XOR function can be better distributed among driver's receive queues in a small number of streams, where it distributes each TCP/UDP stream to a different queue. provides the following option to change the working RSS hash function from Toeplitz to XOR, and vice-versa:

Through sysfs, located at: /sys/class/net/eth*/settings/hfunc.

➤ *To query the operational and supported hash functions:*

```
cat /sys/class/net/eth*/settings/hfunc
```

Example:

```
cat /sys/class/net/eth2/settings/hfunc
Operational hfunc: toeplitz
Supported hfuncs: xor toeplitz
```

➤ *To change the operational hash function:*

```
echo xor > /sys/class/net/eth*/settings/hfunc
```

2.3.1.9.1.1 RSS Verbs Support

Receive Side Scaling (RSS) technology allows spreading incoming traffic between different receive descriptor queues. Assigning each queue to different CPU cores allows better load balancing of the

incoming traffic and improves performance.

This technology was extended to user space by the verbs layer and can be used for RAW ETH QP.

2.3.1.9.1.2 RSS Flow Steering

Steering rules classify incoming packets and deliver a specific traffic type (e.g. TCP/UDP, IP only) or a specific flow to "RX Hash" QP. "RX Hash" QP is responsible for spreading the traffic it handles between the Receive Work Queues using RX hash and Indirection Table. The Receive Work Queue can point to different CQs that can be associated with different CPU cores.

2.3.1.9.1.3 Verbs

The below verbs should be used to achieve this task in both control and data path. Details per verb should be referenced from its man page.

- `ibv_create_wq`, `ibv_modify_wq`, `ibv_destory_wq`
- `ibv_create_rwq_ind_table`, `ibv_destroy_rwq_ind_table`
- `ibv_create_qp_ex` with specific RX configuration to create the "RX hash" QP

2.3.1.10 Time-Stamping

2.3.1.10.1 Time-Stamping Service

Time-stamping is the process of keeping track of the creation of a packet. A time-stamping service supports assertions of proof that a datum existed before a particular time. Incoming packets are time-stamped before they are distributed on the PCI depending on the congestion in the PCI buffers. Outgoing packets are time-stamped very close to placing them on the wire.

2.3.1.10.1.1 Enabling Time-Stamping

Time-stamping is off by default and should be enabled before use.

➤ *To enable time-stamping for a socket:*

Call `setsockopt()` with `SO_TIMESTAMPING` and with the following flags:

<code>SOF_TIMESTAMPING_TX_HARDWARE:</code>	try to obtain send time-stamp in hardware
<code>SOF_TIMESTAMPING_TX_SOFTWARE:</code>	if <code>SOF_TIMESTAMPING_TX_HARDWARE</code> is off or fails, then do it in software
<code>SOF_TIMESTAMPING_RX_HARDWARE:</code>	return the original, unmodified time-stamp as generated by the hardware
<code>SOF_TIMESTAMPING_RX_SOFTWARE:</code>	if <code>SOF_TIMESTAMPING_RX_HARDWARE</code> is off or fails, then do it in software
<code>SOF_TIMESTAMPING_RAW_HARDWARE :</code>	return original raw hardware time-stamp
<code>SOF_TIMESTAMPING_SYS_HARDWARE:</code>	return hardware time-stamp transformed into the system time base
<code>SOF_TIMESTAMPING_SOFTWARE:</code>	return system time-stamp generated in software
<code>SOF_TIMESTAMPING_TX/RX</code>	determine how time-stamps are generated
<code>SOF_TIMESTAMPING_RAW/SYS</code>	determine how they are reported

➤ *To enable time-stamping for a net device:*

Admin privileged user can enable/disable time stamping through calling ioctl (sock, SIOCSHWSTAMP, &ifreq) with the following values:

- Send side time sampling, enabled by ifreq.hwtstamp_config.tx_type when:

```
/* possible values for hwtstamp_config->tx_type */
enum hwtstamp_tx_types {
    /*
     * No outgoing packet will need hardware time stamping;
     * should a packet arrive which asks for it, no hardware
     * time stamping will be done.
     */
    HWTSTAMP_TX_OFF,

    /*
     * Enables hardware time stamping for outgoing packets;
     * the sender of the packet decides which are to be
     * time stamped by setting %SOF_TIMESTAMPING_TX_SOFTWARE
     * before sending the packet.
     */
    HWTSTAMP_TX_ON,

    /*
     * Enables time stamping for outgoing packets just as
     * HWTSTAMP_TX_ON does, but also enables time stamp insertion
     * directly into Sync packets. In this case, transmitted Sync
     * packets will not received a time stamp via the socket error
     * queue.
     */
    HWTSTAMP_TX_ONESTEP_SYNC,
};
Note: for send side time stamping currently only HWTSTAMP_TX_OFF and
HWTSTAMP_TX_ON are supported.
```

- Receive side time sampling, enabled by ifreq.hwtstamp_config.rx_filter when:

```
/* possible values for hwtstamp_config->rx_filter */
enum hwtstamp_rx_filters {
    /* time stamp no incoming packet at all */
    HWTSTAMP_FILTER_NONE,

    /* time stamp any incoming packet */
    HWTSTAMP_FILTER_ALL,
    /* return value: time stamp all packets requested plus some others */
    HWTSTAMP_FILTER_SOME,

    /* PTP v1, UDP, any kind of event packet */
    HWTSTAMP_FILTER_PTP_V1_L4_EVENT,
    /* PTP v1, UDP, Sync packet */
    HWTSTAMP_FILTER_PTP_V1_L4_SYNC,
    /* PTP v1, UDP, Delay_req packet */
    HWTSTAMP_FILTER_PTP_V1_L4_DELAY_REQ,
    /* PTP v2, UDP, any kind of event packet */
    HWTSTAMP_FILTER_PTP_V2_L4_EVENT,
    /* PTP v2, UDP, Sync packet */
    HWTSTAMP_FILTER_PTP_V2_L4_SYNC,
    /* PTP v2, UDP, Delay_req packet */
    HWTSTAMP_FILTER_PTP_V2_L4_DELAY_REQ,

    /* 802.AS1, Ethernet, any kind of event packet */
    HWTSTAMP_FILTER_PTP_V2_L2_EVENT,
    /* 802.AS1, Ethernet, Sync packet */
    HWTSTAMP_FILTER_PTP_V2_L2_SYNC,
    /* 802.AS1, Ethernet, Delay_req packet */
    HWTSTAMP_FILTER_PTP_V2_L2_DELAY_REQ,

    /* PTP v2/802.AS1, any layer, any kind of event packet */
    HWTSTAMP_FILTER_PTP_V2_EVENT,
    /* PTP v2/802.AS1, any layer, Sync packet */
    HWTSTAMP_FILTER_PTP_V2_SYNC,
    /* PTP v2/802.AS1, any layer, Delay_req packet */
    HWTSTAMP_FILTER_PTP_V2_DELAY_REQ,
};
Note: for receive side time stamping currently only HWTSTAMP_FILTER_NONE and
HWTSTAMP_FILTER_ALL are supported.
```

2.3.1.10.1.2 Getting Time-Stamping

Once time stamping is enabled time stamp is placed in the socket Ancillary data. recvmsg() can be used to get this control message for regular incoming packets. For send time stamps the outgoing packet is looped back to the socket's error queue with the send time-stamp(s) attached. It can

be received with `recvmmsg (flags=MSG_ERRQUEUE)`. The call returns the original outgoing packet data including all headers prepended down to and including the link layer, the `scm_time-stamping` control message and a `sock_extended_err` control message with `ee_errno==ENOMSG` and `ee_origin==SO_EE_ORIGIN_TIMESTAMPING`. A socket with such a pending bounced packet is ready for reading as far as `select()` is concerned. If the outgoing packet has to be fragmented, then only the first fragment is time stamped and returned to the sending socket.

When time-stamping is enabled, VLAN stripping is disabled. For more info please refer to [Documentation/networking/timestamping.txt](#) in [kernel.org](#).

On ConnectX-4 and above adapter cards, when time-stamping is enabled, RX CQE compression is disabled (features are mutually exclusive).

2.3.1.10.1.3 Time Stamping Capabilities via ethtool

 To display Time Stamping capabilities via ethtool:

Show Time Stamping capabilities:

```
ethtool -T eth<x>
```

Example:

```
ethtool -T eth0
Time stamping parameters for p2p1:
Capabilities:
                hardware-transmit      (SOF_TIMESTAMPING_TX_HARDWARE)
                software-transmit      (SOF_TIMESTAMPING_TX_SOFTWARE)
                hardware-receive      (SOF_TIMESTAMPING_RX_HARDWARE)
                software-receive      (SOF_TIMESTAMPING_RX_SOFTWARE)
                software-system-clock (SOF_TIMESTAMPING_SOFTWARE)
                hardware-raw-clock    (SOF_TIMESTAMPING_RAW_HARDWARE)

PTP Hardware Clock: 1
Hardware Transmit Timestamp Modes:
off                (HWTSTAMP_TX_OFF)
on                 (HWTSTAMP_TX_ON)

Hardware Receive Filter Modes:
none               (HWTSTAMP_FILTER_NONE)
all                (HWTSTAMP_FILTER_ALL)
```

For more details on PTP Hardware Clock, please refer to: <https://www.kernel.org/doc/Documentation/ptp/ptp.txt>

2.3.1.10.1.4 Steering PTP Traffic to Single RX Ring

As a result of Receive Side Steering (RSS) PTP traffic coming to UDP ports 319 and 320, it may reach the user space application in an out of order manner. In order to prevent this, PTP traffic needs to be steered to single RX ring using ethtool.

Example:

```
# ethtool -u ens7
8 RX rings available
Total 0 rules
# ethtool -U ens7 flow-type udp4 dst-port 319 action 0 loc 1
# ethtool -U ens7 flow-type udp4 dst-port 320 action 0 loc 0
# ethtool -u ens7
```

```

8 RX rings available
Total 2 rules
Filter: 0
Rule Type: UDP over IPv4
Src IP addr: 0.0.0.0 mask: 255.255.255.255
Dest IP addr: 0.0.0.0 mask: 255.255.255.255
TOS: 0x0 mask: 0xff
Src port: 0 mask: 0xffff
Dest port: 320 mask: 0x0
Action: Direct to queue 0
Filter: 1
Rule Type: UDP over IPv4
Src IP addr: 0.0.0.0 mask: 255.255.255.255
Dest IP addr: 0.0.0.0 mask: 255.255.255.255
TOS: 0x0 mask: 0xff
Src port: 0 mask: 0xffff
Dest port: 319 mask: 0x0
Action: Direct to queue 0

```

2.3.1.10.1.5 Tx Port Time-Stamping

Transmitted packet time-stamping accuracy can be improved when using a timestamp generated at the port level instead of a timestamp generated upon CQE creation. Tx port time-stamping better reflects the actual time of a packet's transmission.

Normal Send queues (SQs) are open with CQE time-stamp support. When this feature is enabled, the driver is expected to open extra Tx port time-stamped SQ per traffic class (TC).

The stream must meet the following conditions in order to be transmitted through a Tx port time-stamped SQ.

1. SKBTX_HW_TSTAMP flag was set at tx_flag (SO_TIMESTAMPING was set via setsockopt() or similarly)
2. Packet type is:
 - a. Non-IP, with EtherType of PTP over IEEE 802.3 (0x88f7)
or
 - b. UDP over IPv4/IPv6

This feature is disabled by default in order to avoid extra SQ memory allocations. The feature can be enabled or disabled using the following command.

```
ethtool --set-priv-flags <ifc-name> tx_port_ts on / off
```

2.3.1.10.1.6 PTP Cyc2time Hardware Translation Offload

This feature is supported on ConnectX-6 Dx and above adapter cards only.

Overview

Device timestamp can be in one of two modes: real time or free running internal time.

In free running internal time mode, the device clock is not editable in any way. Driver and/or user space must adjust it to the real-time nanosecond values.

In real time mode, the hardware clock device can be adjusted and can provide timestamps which are already translated into real-time nanoseconds.

Both modes are global per device. Once a mode is set, all clock-related features (such as PPS, CQE TS, PCIe bar, etc) will work with the chosen clock mode only.

Free running internal time is the default mode configured in the hardware. The driver will modify the hardware real time clock based on PTP daemon clock adjustments.

Only physical functions are allowed to modify the hardware real-time clock, so PTP daemon adjustments from VFs will be treated as NOP. In case more than one physical function tries to modify the hardware real-time clock, the device will select one of the functions as its designated clock provider. All other input will also be treated as a NOP. The designated clock provider can be replaced by the device if no new adjustments have been received from the current provider after some period.

Timestamp Format

CQE hardware timestamp format for ConnectX-6 Dx and ConnectX-6 Lx NICs is 64 bit, as follows.

{32bit sec, 32 bit nsec}

Configuration

In order to enable the feature, set `REAL_TIME_CLOCK_ENABLE` in `NV_CONFIG` via `mlxconfig` and restart the driver.

Limitations

- Administrator must restart the driver and perform a FW reset for the configuration to take effect. Otherwise, mismatch between HW and driver timestamp mode might occur.
- Once real time mode is activated on a given device (see configuration section), version 5.3 or newer must run on all device functions. Any older driver running on a device function at this configuration will fail to open any traffic queues (RDMA or ETH), hence becoming dysfunctional.
- In real time mode, all device functions must be PTP-synchronized by a single clock domain—do not use multiple GMs for different functions on the same device.
- Regarding hardware clock ownership, the hardware is configured only from a single elected function; other function settings are ignored by the device. There is no indication as to which function is the hardware-clock's owner. After an internal timeout without modifying the hardware clock, a function loses the hardware-clock's ownership and is open to be grasped by any of the functions.
- All PFs/VFs within the same device must sync to the same 1588 master clock. If multiple masters are used, the device will use a single elected function. This might lead to wrong clock representation by device, wrong 1588 TLVs and hiccups on replacement of elected function.
- This feature is supported on ConnectX-6 Dx and above adapter cards only.

2.3.1.10.2 RoCE Time-Stamping

RoCE Time-Stamping allows you to stamp packets when they are sent to the wire/received from the wire. The time-stamp is given in raw hardware cycles but could be easily converted into hardware referenced nanoseconds based time. Additionally, it enables you to query the hardware for the hardware time, thus stamp other application's event and compare time.

2.3.1.10.2.1 Query Capabilities

Time-stamping is available if and only the hardware reports it is capable of reporting it. To verify whether RoCE Time-Stamping is available, run `ibv_query_device_ex`.

For further information, please see [ibv_query_device_ex manual page](#).

2.3.1.10.2.2 Creating a Time-Stamping Completion Queue

To get time stamps, a suitable extended Completion Queue (CQ) must be created via a special call to `ibv_create_cq_ex` verb.

For further information, please see [ibv_create_cq_ex manual page](#).

Time Stamping is not available when CQE zipping is used.

2.3.1.10.2.3 Querying the Hardware Time

Querying the hardware for time is done via the `ibv_query_rt_values_ex` verb. For example:

For further information, please see [ibv_query_rt_values_ex manual page](#).

2.3.1.10.3 One Pulse Per Second (1PPS)

1PPS is a time synchronization feature that allows the adapter to be able to send or receive 1 pulse per second on a dedicated pin on the adapter card using an SMA connector (SubMiniature version A). Only one pin is supported and could be configured as 1PPS in or 1PPS out.

For further information, refer to [HowTo Test 1PPS on NVIDIA Adapters](#) Community post.

2.3.1.11 Flow Steering

Flow steering is a new model which steers network flows based on flow specifications to specific QPs. Those flows can be either unicast or multicast network flows. In order to maintain flexibility, domains and priorities are used. Flow steering uses a methodology of flow attribute, which is a combination of L2-L4 flow specifications, a destination QP and a priority. Flow steering rules may be inserted either by using `ethtool` or by using InfiniBand verbs. The verbs abstraction uses different terminology from the flow attribute (`ibv_flow_attr`), defined by a combination of specifications (`struct ibv_flow_spec_*`).

2.3.1.11.1 Flow Steering Support

All flow steering features are enabled in the supported adapter cards.

2.3.1.11.2 Flow Domains and Priorities

Flow steering defines the concept of domain and priority. Each domain represents a user agent that can attach a flow. The domains are prioritized. A higher priority domain will always supersede a lower priority domain when their flow specifications overlap. Setting a lower priority value will result in a higher priority.

In addition to the domain, there is a priority within each of the domains. Each domain can have at most 2^{12} priorities in accordance with its needs.

The following are the domains at a descending order of priority:

- User Verbs allows a user application QP to be attached to a specified flow when using `ibv_create_flow` and `ibv_destroy_flow` verbs

- `ibv_create_flow`

```
struct ibv_flow *ibv_create_flow(struct ibv_qp *qp, struct ibv_flow_attr
*flow)
```

Input parameters:

- `struct ibv_qp` - the attached QP.
- `struct ibv_flow_attr` - attaches the QP to the flow specified. The flow contains mandatory control parameters and optional L2, L3 and L4 headers. The optional headers are detected by setting the size and `num_of_specs` fields: `struct ibv_flow_attr` can be followed by the optional flow headers structs:

```
struct ibv_flow_spec_eth
struct ibv_flow_spec_ipv4
struct ibv_flow_spec_tcp_udp
struct ibv_flow_spec_ipv6
```

For further information, please refer to the `ibv_create_flow` man page.

- `ibv_destroy_flow`

```
int ibv_destroy_flow(struct ibv_flow *flow_id)
```

Input parameters:

`ibv_destroy_flow` requires struct `ibv_flow` which is the return value of `ibv_create_flow` in case of success.

Output parameters:

Returns 0 on success, or the value of `errno` on failure.

For further information, please refer to the `ibv_destroy_flow` man page.

2.3.1.11.3 Ethtool

Ethtool domain is used to attach an RX ring, specifically its QP to a specified flow. Please refer to the most recent ethtool man page for all the ways to specify a flow.

Examples:

- `ethtool -U eth5 flow-type ether dst 00:11:22:33:44:55 loc 5 action 2`
All packets that contain the above destination MAC address are to be steered into rx-ring 2 (its underlying QP), with priority 5 (within the ethtool domain)
- `ethtool -U eth5 flow-type tcp4 src-ip 1.2.3.4 dst-port 8888 loc 5 action 2`
All packets that contain the above destination IP address and source port are to be steered into rx- ring 2. When destination MAC is not given, the user's destination MAC is filled automatically.
- `ethtool -U eth5 flow-type ether dst 00:11:22:33:44:55 vlan 45 m 0xf000 loc 5 action 2`
All packets that contain the above destination MAC address and specific VLAN are steered into ring 2. Please pay attention to the VLAN's mask 0xf000. It is required in order to add such a rule.
- `ethtool -u eth5`
Shows all of ethtool's steering rule

When configuring two rules with the same priority, the second rule will overwrite the first one, so this ethtool interface is effectively a table. Inserting Flow Steering rules in the kernel requires support from both the ethtool in the user space and in kernel (v2.6.28).

2.3.1.11.4 Accelerated Receive Flow Steering (aRFS)

Receive Flow Steering (RFS) and Accelerated Receive Flow Steering (aRFS) are kernel features currently available in most distributions. For RFS, packets are forwarded based on the location of the application consuming the packet. aRFS boosts the speed of RFS by adding support for the hardware. By using aRFS (unlike RFS), the packets are directed to a CPU that is local to the thread running the application.

aRFS is an in-kernel logic responsible for load balancing between CPUs by attaching flows to CPUs that are used by flow's owner applications. This domain allows the aRFS mechanism to use the flow steering infrastructure to support the aRFS logic by implementing the `ndo_rx_flow_steer`, which, in turn, calls the underlying flow steering mechanism with the aRFS domain.

➤ To configure RFS:

Configure the RFS flow table entries (globally and per core).

Note: The functionality remains disabled until explicitly configured (by default it is 0).

- The number of entries in the global flow table is set as follows:

```
/proc/sys/net/core/rps_sock_flow_entries
```

- The number of entries in the per-queue flow table are set as follows:

```
/sys/class/net/<dev>/queues/rx-<n>/rps_flow_cnt
```

Example:

```
# echo 32768 > /proc/sys/net/core/rps_sock_flow_entries
# NUM_CHANNELS=`ethtool -l ens6 | grep "Combined:" | tail -1 | awk '{print $2}'`
# for f in `seq 0 $((NUM_CHANNELS-1))`; do echo 32768 > /sys/class/net/ens6/queues/rx-$f/rps_flow_cnt; done
```

➤ To Configure aRFS:

The aRFS feature requires explicit configuration in order to enable it. Enabling the aRFS requires enabling the 'ntuple' flag via the ethtool.

For example, to enable ntuple for eth0, run:

```
ethtool -K eth0 ntuple on
```

aRFS requires the kernel to be compiled with the `CONFIG_RFS_ACCEL` option. This option is available in kernels 2.6.39 and above. Furthermore, aRFS requires Device Managed Flow Steering support.

RFS cannot function if LRO is enabled. LRO can be disabled via ethtool.

2.3.1.11.5 Flow Steering Dump Tool

The `mlx_fs_dump` is a python tool that prints the steering rules in a readable manner. Python v2.7 or above, as well as `pip`, `anytree` and `termcolor` libraries are required to be installed on the host.

Running example:

```
./ofed_scripts/utils/mlx_fs_dump -d /dev/mst/mt4115_pciconf0
FT: 9 (level: 0x18, type: NIC_RX)
+-- FG: 0x15 (MISC)
  |-- FTE: 0x0 (FWD) to (TIR:0x7e) out.ethtype:IPv4 out.ip_prot:UDP out.udp_dport:0x140
  +-- FTE: 0x1 (FWD) to (TIR:0x7e) out.ethtype:IPv4 out.ip_prot:UDP out.udp_dport:0x13f
...
```

For further information on the `mlx_fs_dump` tool, please refer to [mlx_fs_dump](#) Community post.

2.3.1.12 Wake-on-LAN (WoL)

Wake-on-LAN (WoL) is a technology that allows a network professional to remotely power on a computer or to wake it up from sleep mode.

- To enable WoL:

```
ethtool -s <interface> wol g
```

- To get WoL:

```
ethtool <interface> | grep Wake-on Wake-on: g
```

Where:

"g" is the magic packet activity.

2.3.1.13 Hardware Accelerated 802.1ad VLAN (Q-in-Q Tunneling)

Q-in-Q tunneling allows the user to create a Layer 2 Ethernet connection between two servers. The user can segregate a different VLAN traffic on a link or bundle different VLANs into a single VLAN. Q-in-Q tunneling adds a service VLAN tag before the user's 802.1Q VLAN tags.

For Q-in-Q support in virtualized environments (SR-IOV), please refer to ["Q-in-Q Encapsulation per VF in Linux \(VST\)"](#).

➤ To enable device support for accelerated 802.1ad VLAN:

1. Turn on the new ethtool private flag "phv-bit" (disabled by default).

```
$ ethtool --set-priv-flags eth1 phv-bit on
```

Enabling this flag sets the `phv_en` port capability.

2. Change the interface device features by turning on the ethtool device feature "tx-vlan- stag-hw-insert" (disabled by default).

```
$ ethtool -K eth1 tx-vlan-stag-hw-insert on
```

Once the private flag and the ethtool device feature are set, the device will be ready for 802.1ad VLAN acceleration.

The "phv-bit" private flag setting is available for the Physical Function (PF) only. The Virtual Function (VF) can use the VLAN acceleration by setting the "tx-vlan-stag-hw-insert" parameter only if the private flag "phv-bit" is enabled by the PF. If the PF enables/disables the "phv-bit" flag after the VF driver is up, the configuration will take place only after the VF driver is restarted.

2.3.1.14 VLAN Stripping in Linux Verbs

This capability is now accessible from userspace using the verbs.

VLAN stripping adds access to the device's ability to offload the Customer VLAN (cVLAN) header stripping from an incoming packet, thus achieving acceleration of VLAN handing in receive flow.

It is configured per WQ option. You can either enable it upon creation or modify it later using the appropriate verbs (`ibv_create_wq` / `ibv_modify_wq`).

2.3.1.15 Offloaded Traffic Sniffer

To be able to activate this feature, make sure libpcap library v1.9 or above is installed on your setup.

To download libpcap, please visit <https://www.tcpdump.org/>.

Offloaded Traffic Sniffer allows bypass kernel traffic (such as RoCE, VMA, and DPDK) to be captured by existing packet analyzer, such as tcpdump.

To capture the interface's bypass kernel traffic, run tcpdump on the RDMA device.

For examples on how to dump RDMA traffic using the Inbox tcpdump tool for ConnectX-4 adapter cards and above, click [here](#).

Note that enabling Offloaded Traffic Sniffer can cause bypass kernel traffic speed degradation.

In case you do not wish to install libpcap on your setup, you can use docker to run the tcpdump. For further information, please see <https://hub.docker.com/r/mellanox/tcpdump-rdma>.

2.3.1.16 Dump Configuration

This feature helps dumping driver and firmware configuration using ethtool. It creates a backup of the configuration files into a specified dump file.

2.3.1.16.1 Dump Parameters (Bitmap Flag)

The following bitmap parameters are used to set the type of dump:

Bitmap Parameters

Value	Description
1	MST dump
2	Ring dump (Software context information for SQs, EQs, RQs, CQs)
3	MST dump + Ring dump (1+2)
4	Clear this parameter

2.3.1.16.2 Configuration

In order to configure this feature, follow the steps below:

1. Set the dump bitmap parameter by running `-W` (uppercase) with the desired bitmap parameter value (see Bitmap Parameters table above). In the following example, the bitmap parameter value is 3.

```
ethtool -W ens1f0 3
```

2. Dump the file by running `-w` (lowercase) with the desired configuration file name.

```
ethtool -w ens1f0 data /tmp/dump.bin
```

3. [Optional] To get the bitmap parameter value, version and size of the dump, run the command above without the file name.

```
ethtool -w ens1f0  
flag: 3, version: 1, length: 4312
```

4. To open the dump file, run:

```
mlnx_dump_parser -f /tmp/dump.bin -m mst_dump_demo.txt -r ring_dump_demo.txt  
Version: 1 Flag: 3 Number of blocks: 123 Length 327584  
MCION module number: 0 status: | present |  
DRIVER VERSION: 1-23 (03 Mar 2015)  
DEVICE NAME 0000:81:00.0:ens1f0  
Parsing Complete!
```

where:

-f	For the file to be parsed (the file that was just created)
-m	For the mst dump file
-r	For the ring dump file

For further information, refer to [HowTo Dump Driver Configuration \(via ethtool\)](#) Community post.

Output:

```
# mlnx_dump_parser -f /tmp/dump.bin -m mst_dump_demo.txt -r ring_dump_demo.txt
Version: 1 Flag: 3 Number of blocks: 123 Length 327584
MCION module number: 0 status: | present |
DRIVER VERSION: 1-23 (03 Mar 2015)
DEVICE NAME 0000:81:00.0:ens1f0
Parsing Complete!
```

5. Open the files.

- a. The MST dump file will look as follows. In order to analyze it, contact [NVIDIA Support](#).

```
cat mst_dump_demo.txt
0x00000000 0x01002000
0x00000004 0x00000000
0x00000008 0x00000000
0x0000000c 0x00000000
0x00000010 0x00000000
0x00000014 0x00000000
0x00000018 0x00000000
...
```

- b. The Ring dump file can help developers debug ring-related issues, and it looks as follows:

```
# cat ring_dump_demo.txt
SQ TYPE: 3, WQN: 102, PI: 0, CI: 0, STRIDE: 6, SIZE: 1024...
SQ TYPE: 3, WQN: 102, PI: 0, CI: 0, STRIDE: 6, SIZE: 1024, WQE_NUM: 65536, GROUP_IP: 0
CQ TYPE: 5, WQN: 20, PI: 0, CI: 0, STRIDE: 6, SIZE: 1024, WQE_NUM: 1024, GROUP_IP: 0
RQ TYPE: 4, WQN: 103, PI: 15, CI: 0, STRIDE: 5, SIZE: 16, WQE_NUM: 512, GROUP_IP: 0
CQ TYPE: 5, WQN: 21, PI: 0, CI: 0, STRIDE: 6, SIZE: 16384, WQE_NUM: 16384, GROUP_IP: 0
EQ TYPE: 6, CI: 1, SIZE: 0, IRQN: 109, EQN: 19, NENT: 2048, MASK: 0, INDEX: 0, GROUP_ID: 0
SQ TYPE: 3, WQN: 106, PI: 0, CI: 0, STRIDE: 6, SIZE: 1024, WQE_NUM: 65536, GROUP_IP: 1
CQ TYPE: 5, WQN: 23, PI: 0, CI: 0, STRIDE: 6, SIZE: 1024, WQE_NUM: 1024, GROUP_IP: 1
RQ TYPE: 4, WQN: 107, PI: 15, CI: 0, STRIDE: 5, SIZE: 16, WQE_NUM: 512, GROUP_IP: 1
CQ TYPE: 5, WQN: 24, PI: 0, CI: 0, STRIDE: 6, SIZE: 16384, WQE_NUM: 16384, GROUP_IP: 1
EQ TYPE: 6, CI: 1, SIZE: 0, IRQN: 110, EQN: 20, NENT: 2048, MASK: 0, INDEX: 1, GROUP_ID: 1
SQ TYPE: 3, WQN: 110, PI: 0, CI: 0, STRIDE: 6, SIZE: 1024, WQE_NUM: 65536, GROUP_IP: 2
CQ TYPE: 5, WQN: 26, PI: 0, CI: 0, STRIDE: 6, SIZE: 1024, WQE_NUM: 1024, GROUP_IP: 2
RQ TYPE: 4, WQN: 111, PI: 15, CI: 0, STRIDE: 5, SIZE: 16, WQE_NUM: 512, GROUP_IP: 2
CQ TYPE: 5, WQN: 27, PI: 0, CI: 0, STRIDE: 6, SIZE: 16384, WQE_NUM: 16384, GROUP_IP: 2
...
```

2.3.1.17 Local Loopback Disable

Local Loopback Disable feature allows users to force the disablement of local loopback on the virtual port (vport). This disables both unicast and multicast loopback in the hardware.

- To enable Local Loopback Disable, run the following command:

```
echo 1 > /sys/class/net/<iface>/settings/force_local_lb_disable"
```

- To disable Local Loopback Disable, run the following command:

```
echo 0 > /sys/class/net/<iface>/settings/force_local_lb_disable"
```

When turned off, the driver configures the loopback mode according to its own logic.

2.3.1.18 Kernel Transport Layer Security (kTLS) Offloads

This feature is supported on ConnectX-6 Dx crypto cards only.

2.3.1.18.1 Overview

Transport Layer Security (TLS) is a widely-deployed protocol used for securing TCP connections on the Internet. TLS is also a required feature for HTTP/2, the latest web standard. Kernel implementation of TLS (kTLS) provides new opportunities for offloading the protocol into the hardware.

TLS data-path offload allows the NIC to accelerate encryption, decryption and authentication of AES-GCM. TLS offload handles data as it goes through the device without storing any data, but only updating context. If the packet cannot be encrypted/decrypted by the device, then a software fallback handles the packet.

2.3.1.18.2 Establishing a kTLS Connection

To avoid unnecessary complexity in the kernel, the TLS handshake is kept in the user space. A full TLS connection using the socket is done using the following scheme:

1. Call `connect()` or `accept()` on a standard TCP file descriptor.
2. Use a user space TLS library to complete a handshake.
3. Create a new kTLS socket file descriptor.
4. Extract the TLS Initialization Vectors (IVs), session keys, and sequence IDs from the TLS library. Use the `setsockopt` function on the kTLS file descriptor (FD) to pass them to the kernel.
5. Use standard `read()`, `write()`, `sendfile()` and `splice()` system calls on the kTLS FD.

Drivers can offer Tx and Rx packet encryption/decryption offload from the kernel into the NIC hardware. Upon receipt of a non-data TLS message (a control message), the kTLS socket returns an error, and the message is left on the original TCP socket instead. The kTLS socket is automatically unattached. Transfer of control back to the original encrypted FD is done by calling `getsockopt` to receive the current sequence numbers, and inserting them into the TLS library.

2.3.1.18.3 Kernel Support

For support in the kernel, make sure the following flags are set as follows.

- `CONFIG_TLS=y`
- `CONFIG_TLS_DEVICE=y | m`

For kTLS Tx device offloads with OFED drivers, kernel TLS module (`kernel/net/tls`) must be aligned to kernel v5.3 and above.

For kTLS Rx device offloads with OFED drivers, kernel TLS module (`kernel/net/tls`) must be aligned to kernel v5.9 and above.

2.3.1.18.4 Configuring kTLS Offloads

➤ To enable kTLS Tx offload, run:

```
ethtool -K <if> tls-hw-tx-offload on
```

➤ To enable kTLS Rx offload, run:

```
ethtool -K <if> tls-hw-rx-offload on
```

For further information on TLS offloads, please visit the following kernel documentation:

- <https://www.kernel.org/doc/html/latest/networking/tls-offload.html>
- <https://www.kernel.org/doc/html/latest/networking/tls.html#kernel-tls>

2.3.1.18.5 OpenSSL with kTLS Offload

OpenSSL version 3.0.0 or above is required to support kTLS TX/RX offloads.

Supported OpenSSL version is available to download from distro packages, or can be downloaded and compiled from the OpenSSL github.

2.3.1.19 IPsec Crypto Offload

This feature is supported on crypto-enabled products of BlueField-2 DPUs, and on ConnectX-6 Dx, ConnectX-6 Lx and ConnectX-7 adapters. Newer/future crypto-enabled DPU and adapter product generations should also support the feature, unless explicitly stated in their documentation.

For NVIDIA BlueField-2 DPUs and ConnectX-6 Dx adapters Only: If your target application will utilize bandwidth of 100Gb/s or higher, where a substantial part of the bandwidth will be allocated for IPsec traffic, please refer to the NVIDIA BlueField-2 DPUs Product Release Notes or NVIDIA ConnectX-6 Dx Adapters Product Release Notes document to learn about a potential bandwidth limitation. To access the relevant product release notes, please contact your NVIDIA sales representative.

Overview and Configuration

IPsec crypto offload feature, also known as IPsec inline offload or IPsec aware offload feature enables the user to offload IPsec crypto encryption and decryption operations to the hardware.

Note that the hardware implementation only supports AES-GCM encryption scheme.

To enable the feature, support in both kernel and adapter firmware is required.

- For support in the kernel, make sure the following flags are set as follows.


```
CONFIG_XFRM_OFFLOAD=y
CONFIG_INET_ESP_OFFLOAD=m
CONFIG_INET6_ESP_OFFLOAD=m
```

Note: These flags are enabled by default in RedHat 8 and Ubuntu 18.04.

- For support in the firmware, make sure the below string is found in the dmesg.

```
mlx5e: IPsec ESP acceleration enabled
```

2.3.1.19.1 Configuring Security Associations for IPsec Offloads

To program the inline offload security associations (SA), add the option "offload dev <netdev interface> dir out/in" in the "ip xfrm state" command for transmitting and receiving SA.

Transmit inline offload SA xfrm command example:

```
sudo ip xfrm state add src 192.168.1.64/24 dst 192.168.1.65/24 proto esp spi 0x46dc6204 reqid 0x46dc6204 mode transport aead 'rfc4106(gcm(aes))' 0x60bd6c3eafba371a46411830fd56c53af93883261ed1fb26767820ff493f43ba35b0dcca 128 offload dev p4pl dir out sel src 192.168.1.64 dst 192.168.1.65
```

Receive inline offload SA xfrm command example:

```
sudo ip xfrm state add src 192.168.1.65/24 dst 192.168.1.64/24 proto esp spi 0xaea0846c reqid 0xaea0846c mode transport aead 'rfc4106(gcm(aes))' 0x81d5c3167c912c1dd50dab0cb4b6d815b6ace8844304db362215a258cd19deda8f89deda 128 offload dev p4pl dir in sel src 192.168.1.65 dst 192.168.1.64
```

2.3.1.19.1.1 Setting xfrm Policies Example

First server:

```
+ sudo ip xfrm state add src 192.168.1.64/24 dst 192.168.1.65/24 proto esp spi 0x28f39549 reqid 0x28f39549 mode transport aead 'rfc4106(gcm(aes))' 0x492e8ffe718a95a00c1893ea61afc64997f4732848ccfe6ea07db483175cb18de9ae411a 128 offload dev enp4s0 dir out sel src 192.168.1.64 dst 192.168.1.65
+ sudo ip xfrm state add src 192.168.1.65/24 dst 192.168.1.64/24 proto esp spi 0x622a73b4 reqid 0x622a73b4 mode transport aead 'rfc4106(gcm(aes))' 0x093bfee2212802d626716815f862da31bcc7d9c44cfe3ab8049e7604b2feb1254869d25b 128 offload dev enp4s0 dir in sel src 192.168.1.65 dst 192.168.1.64
+ sudo ip xfrm policy add src 192.168.1.64 dst 192.168.1.65 dir out tmpl src 192.168.1.64/24 dst 192.168.1.65/24 proto esp reqid 0x28f39549 mode transport
+ sudo ip xfrm policy add src 192.168.1.65 dst 192.168.1.64 dir in tmpl src 192.168.1.65/24 dst 192.168.1.64/24 proto esp reqid 0x622a73b4 mode transport
+ sudo ip xfrm policy add src 192.168.1.65 dst 192.168.1.64 dir fwd tmpl src 192.168.1.65/24 dst 192.168.1.64/24 proto esp reqid 0x622a73b4 mode transport
```

Second server:

```
+ ssh -A -t root@1-csi-0921d /bin/bash
+ set -e
+ '[' 0 == 1 ']'
+ sudo ip xfrm state add src 192.168.1.64/24 dst 192.168.1.65/24 proto esp spi 0x28f39549 reqid 0x28f39549 mode transport aead 'rfc4106(gcm(aes))' 0x492e8ffe718a95a00c1893ea61afc64997f4732848ccfe6ea07db483175cb18de9ae411a 128 offload dev enp4s0 dir in sel src 192.168.1.64 dst 192.168.1.65
+ sudo ip xfrm state add src 192.168.1.65/24 dst 192.168.1.64/24 proto esp spi 0x622a73b4 reqid 0x622a73b4 mode transport aead 'rfc4106(gcm(aes))' 0x093bfee2212802d626716815f862da31bcc7d9c44cfe3ab8049e7604b2feb1254869d25b 128 offload dev enp4s0 dir out sel src 192.168.1.65 dst 192.168.1.64
+ sudo ip xfrm policy add src 192.168.1.65 dst 192.168.1.64 dir out tmpl src 192.168.1.65/24 dst 192.168.1.64/24 proto esp reqid 0x622a73b4 mode transport
+ sudo ip xfrm policy add src 192.168.1.64 dst 192.168.1.65 dir in tmpl src 192.168.1.64/24 dst 192.168.1.65/24 proto esp reqid 0x28f39549 mode transport
+ sudo ip xfrm policy add src 192.168.1.64 dst 192.168.1.65 dir fwd tmpl src 192.168.1.64/24 dst 192.168.1.65/24 proto esp reqid 0x28f39549 mode transport
+ echo 'IPsec tunnel configured successfully'
```

2.3.1.20 MACsec Full Offload

MACsec Full offload feature, also known as MACsec inline Full offload, enables the user to offload MACsec crypto encryption and decryption, MACsec headers encapsulation and decapsulation, and Anti replay operations to the hardware.

Hardware implementation supports GCM-AES & GCM-AES-XPB encryption schemes and is supported with ConnectX-7 onwards.

To enable the feature, support in both kernel and adapter firmware is required.

For support in the kernel, make sure the following flags are set as follows:

- CONFIG_MACSEC=y
- CONFIG_MLX5_EN_MACSEC=y

For support in firmware use the following version:

- xx.34.0364 and up

2.3.1.20.1 Configurations

2.3.1.20.1.1 IProute2 Configuration

Configuring Physical Interface

Client side:

- ip address flush <physical_device>
- ip address add <client_physical_device_ip> dev <physical interface>
- ip link set dev <physical_device>up

Server side:

- ip address flush <physical_device>
- ip address add <server_physical_device_ip> dev <physical interface>
- ip link set dev <physical_device>up

Add MACsec Device

Client side:

- ip link add link <physical_device> <macsec_device> type macsec sci <client_sci> client on

Server side:

- ip link add link <physical_device> <macsec_device> type macsec sci <client_sci> client on

Offload MACsec Device

Client side:

- ip macsec offload <macsec_device> mac

Server side:

- ip macsec offload <macsec_device> mac

Add MACsec rules:

Client side:

- ip macsec add <macsec_device> tx sa <sa_num>pn <inital_packet_number>on key <client_key_id> <client_key>
- ip macsec add <macsec_device> rx sci <server_sci> on
- ip macsec add <macsec_device> rx sci <server_sci>sa <sa_num> pn <inital_packet_number> on key <server_key_id> <server_key>

Server side:

- ip macsec add <macsec_device> tx sa <sa_num>pn <inital_packet_number>on key <server_key_id> <server_key>
- ip macsec add <macsec_device> rx sci <client_sci> on
- ip macsec add <macsec_device> rx sci <client_sci>sa <sa_num> pn <inital_packet_number> on key <client_key_id> <client_key>

Configure MACsec Device IPs:

Client side:

- ip address flush <macsec_device>
- ip address add <client_macsec_device_ip> dev <macsec_device>
- ip link set dev <macsec_device> up

Server side:

- ip address flush <macsec_device>
- ip address add <server_macsec_device_ip> dev <macsec_device>
- ip link set dev <macsec_device> up

2.3.1.20.1.2 Configuration Example

Client side:

- ip address flush enp8s0f0
- ip address add 1.1.1.1/24 dev enp8s0f0
- ip link set dev enp8s0f0 up
- ip link add link enp8s0f0 macsec0 type macsec sci 1 encrypt on
- ip macsec offload macsec0 mac
- ip macsec add macsec0 tx sa 0 pn 1 on key 00 dffafc8d7b9a43d5b9a3dfbbf6a30c16
- ip macsec add macsec0 rx sci 2 on
- ip macsec add macsec0 rx sci 2 sa 0 pn 1 on key 00 ead3664f508eb06c40ac7104cdae4ce5
- ip address flush macsec0
- ip address add 2.2.2.1/24 dev macsec0
- ip link set dev macsec0 up

Server side:

- ip link del macsec0
- ip address flush enp8s0f0
- ip address add 1.1.1.2/24 dev enp8s0f0
- ip link set dev enp8s0f0 up
- ip link add link enp8s0f0 macsec0 type macsec sci 2 encrypt on
- ip macsec offload macsec0 mac
- ip macsec add macsec0 tx sa 0 pn 1 on key 00 ead3664f508eb06c40ac7104cdae4ce5
- ip macsec add macsec0 rx sci 1 on
- ip macsec add macsec0 rx sci 1 sa 0 pn 1 on key 00 dffafc8d7b9a43d5b9a3dfbbf6a30c16
- ip address flush macsec0
- ip address add 2.2.2.2/24 dev macsec0
- ip link set dev macsec0 up

- Use: "ip macsec show" command to check configuration
- To make sure traffic is offloaded, check MACsec counters: `ethtool -S <physical_device> | grep macsec "`

Additional Resources

Linux Manual page: [linux_manual](#)

2.3.2 Virtualization

The chapter contains the following sections:

- [Single Root IO Virtualization \(SR-IOV\)](#)
- [Enabling Paravirtualization](#)
- [VXLAN Hardware Stateless Offloads](#)
- [Q-in-Q Encapsulation per VF in Linux \(VST\)](#)
- [802.1Q Double-Tagging](#)

2.3.2.1 Single Root IO Virtualization (SR-IOV)

Single Root IO Virtualization (SR-IOV) is a technology that allows a physical PCIe device to present itself multiple times through the PCIe bus. This technology enables multiple virtual instances of the device with separate resources. NVIDIA adapters are capable of exposing up to 127 virtual instances (Virtual Functions (VFs) for each port in the NVIDIA ConnectX® family cards. These virtual functions can then be provisioned separately. Each VF can be seen as an additional device connected to the Physical Function. It shares the same resources with the Physical Function, and its number of ports equals those of the Physical Function.

SR-IOV is commonly used in conjunction with an SR-IOV enabled hypervisor to provide virtual machines direct hardware access to network resources hence increasing its performance.

In this chapter we will demonstrate setup and configuration of SR-IOV in a Red Hat Linux environment using ConnectX® VPI adapter cards.

2.3.2.1.1 System Requirements

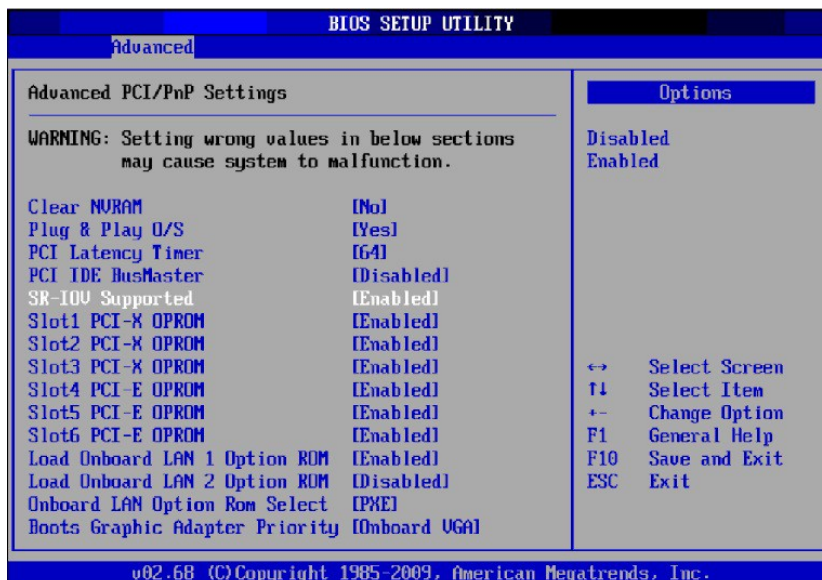
To set up an SR-IOV environment, the following is required:

- MLNX_EN Driver
- A server/blade with an SR-IOV-capable motherboard BIOS
- Hypervisor that supports SR-IOV such as: Red Hat Enterprise Linux Server Version 6
- NVIDIA ConnectX® VPI Adapter Card family with SR-IOV capability

2.3.2.1.2 Setting Up SR-IOV

Depending on your system, perform the steps below to set up your BIOS. The figures used in this section are for illustration purposes only. For further information, please refer to the appropriate BIOS User Manual:

1. Enable "SR-IOV" in the system BIOS.



2. Enable "Intel Virtualization Technology".



3. Install a hypervisor that supports SR-IOV.
4. Depending on your system, update the `/boot/grub/grub.conf` file to include a similar command line load parameter for the Linux kernel.
For example, to Intel systems, add:

```
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Red Hat Enterprise Linux Server (4.x.x)
  root (hd0,0)
  kernel /vmlinuz-4.x.x ro root=/dev/VolGroup00/LogVol100 rhgb quiet
  intel_iommu=on          initrd /initrd-4.x.x.img
```

Note: Please make sure the parameter "`intel_iommu=on`" exists when updating the `/boot/grub/grub.conf` file, otherwise SR-IOV cannot be loaded.

Some OSs use `/boot/grub2/grub.cfg` file. If your server uses such file, please edit this file instead (add "`intel_iommu=on`" for the relevant menu entry at the end of the line that starts with "linux16").

2.3.2.1.3 Configuring SR-IOV (Ethernet)

To set SR-IOV in Ethernet mode, refer to [HowTo Configure SR-IOV for ConnectX-4/ConnectX- 5/ ConnectX-6 with KVM \(Ethernet\)](#) Community Post.

2.3.2.1.4 Additional SR-IOV Configurations

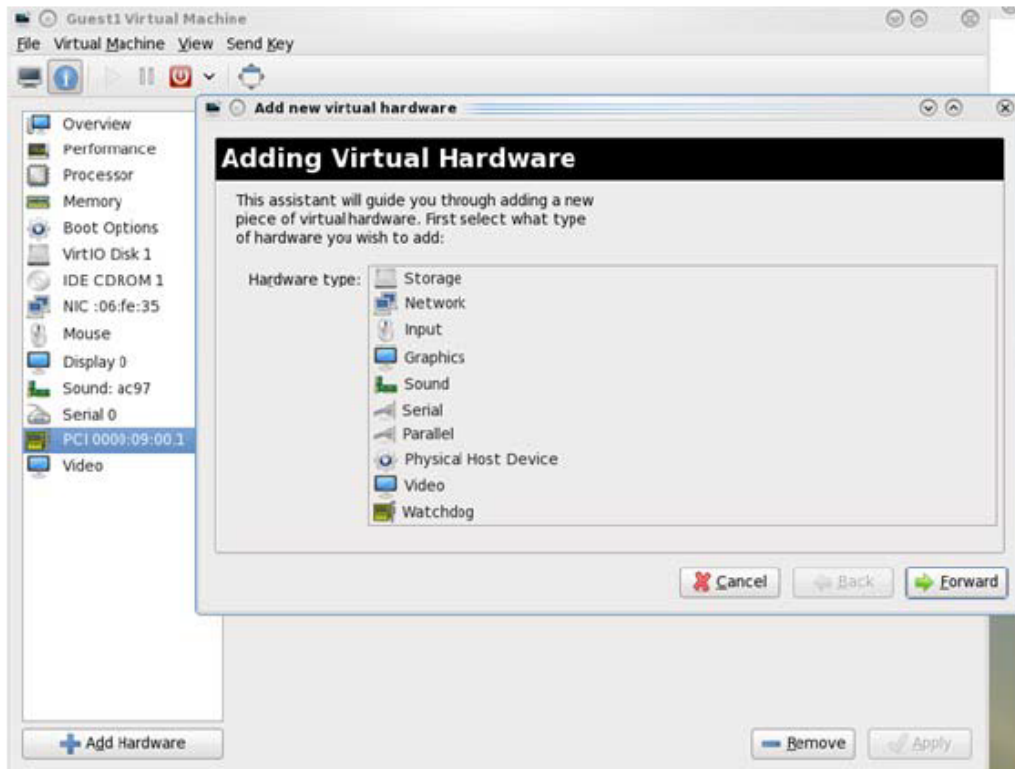
2.3.2.1.4.1 Assigning a Virtual Function to a Virtual Machine

This section describes a mechanism for adding a SR-IOV VF to a Virtual Machine.

2.3.2.1.4.2 Assigning the SR-IOV Virtual Function to the Red Hat KVM VM Server

1. Run the `virt-manager`.

2. Double click on the virtual machine and open its Properties.
3. Go to Details → Add hardware → PCI host device.



4. Choose a NVIDIA virtual function according to its PCI device (e.g., 00:03.1)
5. If the Virtual Machine is up reboot it, otherwise start it.
6. Log into the virtual machine and verify that it recognizes the NVIDIA card. Run:

```
lspci | grep Mellanox
```

Example:

```
lspci | grep Mellanox
01:00.0 Infiniband controller: Mellanox Technologies MT28800 Family [ConnectX-5 Ex]
```

7. Add the device to the `/etc/sysconfig/network-scripts/ifcfg-ethX` configuration file. The MAC address for every virtual function is configured randomly, therefore it is not necessary to add it.

2.3.2.1.4.3 Ethernet Virtual Function Configuration when Running SR-IOV

SR-IOV Virtual function configuration can be done through Hypervisor iprout2/netlink tool, if present. Otherwise, it can be done via sysfs.

```
ip link set { dev DEVICE | group DEVGROUP } [ { up | down } ]
...
[ vf NUM [ mac LLADDR ] [ vlan VLANID [ qos VLAN-QOS ] ]
...
[ spoofchk { on | off } ] ]
...
sysfs configuration (ConnectX-4):
```

```

/sys/class/net/enp8s0f0/device/sriov/[VF]
+-- [VF]
| +-- config
| +-- link_state
| +-- mac
| +-- mac_list
| +-- max_tx_rate
| +-- min_tx_rate
| +-- spoofcheck
| +-- stats
| +-- trunk
| +-- trust
| +-- vlan

```

VLAN Guest Tagging (VGT) and VLAN Switch Tagging (VST)

When running ETH ports on VGT, the ports may be configured to simply pass through packets as is from VFs (VLAN Guest Tagging), or the administrator may configure the Hypervisor to silently force packets to be associated with a VLAN/Qos (VLAN Switch Tagging).

In the latter case, untagged or priority-tagged outgoing packets from the guest will have the VLAN tag inserted, and incoming packets will have the VLAN tag removed.

The default behavior is VGT.

To configure VF VST mode, run:

```
ip link set dev <PF device> vf <NUM> vlan <vlan_id> [qos <qos>]
```

where:

- NUM = 0..max-vf-num
- vlan_id = 0..4095
- qos = 0..7

For example:

- ip link set dev eth2 vf 2 vlan 10 qos 3 - sets VST mode for VF #2 belonging to PF eth2, with vlan_id = 10 and qos = 3
- ip link set dev eth2 vf 2 vlan 0 - sets mode for VF 2 back to VGT

Additional Ethernet VF Configuration Options

- Guest MAC configuration - by default, guest MAC addresses are configured to be all zeroes. If the administrator wishes the guest to always start up with the same MAC, he/she should configure guest MACs before the guest driver comes up. The guest MAC may be configured by using:

```
ip link set dev <PF device> vf <NUM> mac <LLADDR>
```

For legacy and ConnectX-4 guests, which do not generate random MACs, the administrator should always configure their MAC addresses via IP link, as above.

- Spoof checking - Spoof checking is currently available only on upstream kernels newer than 3.1.

```
ip link set dev <PF device> vf <NUM> spoofchk [on | off]
```

- Guest Link State


```
ip link set dev <PF device> vf <UM> state [enable| disable| auto]
```

Virtual Function Statistics

Virtual function statistics can be queried via sysfs:

```
cat /sys/class/infiniband/mlx5_2/device/sriov/2/stats tx_packets : 5011
tx_bytes : 4450870
tx_dropped : 0
rx_packets : 5003
rx_bytes : 4450222
rx_broadcast : 0
rx_multicast : 0
tx_broadcast : 0
tx_multicast : 8
rx_dropped : 0
```

Mapping VFs to Ports

➤ To view the VFs mapping to ports:

Use the ip link tool v2.6.34-3 and above.

```
ip link
```

Output:

```
61: p1p1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen 1000
    link/ether 00:02:c9:f1:72:e0 brd ff:ff:ff:ff:ff:ff
    vf 0 MAC 00:00:00:00:00:00, vlan 4095, spoof checking off, link-state auto
    vf 37 MAC 00:00:00:00:00:00, vlan 4095, spoof checking off, link-state auto
    vf 38 MAC ff:ff:ff:ff:ff:ff, vlan 65535, spoof checking off, link-state disable
    vf 39 MAC ff:ff:ff:ff:ff:ff, vlan 65535, spoof checking off, link-state disable
```

When a MAC is ff:ff:ff:ff:ff:ff, the VF is not assigned to the port of the net device it is listed under. In the example above, vf38 is not assigned to the same port as p1p1, in contrast to vf0. However, even VFs that are not assigned to the net device, could be used to set and change its settings. For example, the following is a valid command to change the spoof check:

```
ip link set dev p1p1 vf 38 spoofchk on
```

This command will affect only the vf38. The changes can be seen in ip link on the net device that this device is assigned to.

RoCE Support

RoCE is supported on Virtual Functions and VLANs may be used with it. For RoCE, the hypervisor GID table size is of 16 entries while the VFs share the remaining 112 entries. When the number of VFs is larger than 56 entries, some of them will have GID table with only a single entry which is inadequate if VF's Ethernet device is assigned with an IP address.

2.3.2.1.4.4 Virtual Guest Tagging (VGT+)

VGT+ is an advanced mode of Virtual Guest Tagging (VGT), in which a VF is allowed to tag its own packets as in VGT, but is still subject to an administrative VLAN trunk policy. The policy determines which VLAN IDs are allowed to be transmitted or received. The policy does not determine the user priority, which is left unchanged.

Packets can be sent in one of the following modes: when the VF is allowed to send/receive untagged and priority tagged traffic and when it is not. No default VLAN is defined for VGT+ port. The send packets are passed to the eSwitch only if they match the set, and the received packets are forwarded to the VF only if they match the set.

Configuration

When working in SR-IOV, the default operating mode is VGT.

➤ *To enable VGT+ mode:*

Set the corresponding port/VF (in the example below port eth5, VF0) range of allowed VLANs.

```
echo "<add> <start_vid> <end_vid>" > /sys/class/net/eth5/device/sriov/0/trunk
```

Examples:

- Adding VLAN ID range (4-15) to trunk:

```
echo add 4 15 > /sys/class/net/eth5/device/sriov/0/trunk
```

- Adding a single VLAN ID to trunk:

```
echo add 17 17 > /sys/class/net/eth5/device/sriov/0/trunk
```

Note: When VLAN ID = 0, it indicates that untagged and priority-tagged traffics are allowed

➤ *To disable VGT+ mode, make sure to remove all VLANs.*

```
echo rem 0 4095 > /sys/class/net/eth5/device/sriov/0/trunk
```

➤ *To remove selected VLANs.*

- Remove VLAN ID range (4-15) from trunk:

```
echo rem 4 15 > /sys/class/net/eth5/device/sriov/0/trunk
```

- Remove a single VLAN ID from trunk:

```
echo rem 17 17 > /sys/class/net/eth5/device/sriov/0/trunk
```

2.3.2.1.4.5 SR-IOV Advanced Security Features

SR-IOV MAC Anti-Spoofing

Normally, MAC addresses are unique identifiers assigned to network interfaces, and they are fixed addresses that cannot be changed. MAC address spoofing is a technique for altering the MAC address to serve different purposes. Some of the cases in which a MAC address is altered can be legal, while others can be illegal and abuse security mechanisms or disguises a possible attacker.

The SR-IOV MAC address anti-spoofing feature, also known as MAC Spoof Check provides protection against malicious VM MAC address forging. If the network administrator assigns a MAC address to a VF (through the hypervisor) and enables spoof check on it, this will limit the end user to send traffic only from the assigned MAC address of that VF.

MAC Anti-Spoofing Configuration

MAC anti-spoofing is disabled by default.

In the configuration example below, the VM is located on VF-0 and has the following MAC address: 11:22:33:44:55:66.

There are two ways to enable or disable MAC anti-spoofing:

1. Use the standard IP link commands - available from Kernel 3.10 and above.
 - a. To enable MAC anti-spoofing, run:

```
ip link set ens785f1 vf 0 spoofchk on
```

- b. To disable MAC anti-spoofing, run:

```
ip link set ens785f1 vf 0 spoofchk off
```

2. Specify echo "ON" or "OFF" to the file located under `/sys/class/net/<ifname / device/sriov/<VF index>/spoofcheck`.

- a. To enable MAC anti-spoofing, run:

```
echo "ON" > /sys/class/net/ens785f1/vf/0/spoofchk
```

- b. To disable MAC anti-spoofing, run:

```
echo "OFF" > /sys/class/net/ens785f1/vf/0/spoofchk
```

This configuration is non-persistent and does not survive driver restart.

Limit and Bandwidth Share Per VF

This feature enables rate limiting traffic per VF in SR-IOV mode. For details on how to configure rate limit per VF for ConnectX-4 and above adapter cards, please refer to [HowTo Configure Rate Limit per VF for ConnectX-4/ConnectX-5/ConnectX-6](#) Community post.

Limit Bandwidth per Group of VFs

VFs Rate Limit for vSwitch (OVS) feature allows users to join available VFs into groups and set a rate limitation on each group. Rate limitation on a VF group ensures that the total Tx bandwidth that the VFs in this group get (altogether combined) will not exceed the given value.

With this feature, a VF can still be configured with an individual rate limit as in the past (under `/sys/class/net/<ifname>/device/sriov/<vf_num>/max_tx_rate`). However, the actual bandwidth limit on the VF will eventually be determined considering the VF group limitation and how many VFs are

in the same group.

For example: 2 VFs (0 and 1) are attached to group 3.

Case 1: The rate limitation on the group is set to 20G. Rate limit of each VF is 15G

Result: Each VF will have a rate limit of 10G

Case 2: Group's max rate limitation is still set to 20G. VF 0 is configured to 30G limit, while VF 1 is configured to 5G rate limit

Result: VF 0 will have 15G de-facto. VF 1 will have 5G

The rule of thumb is that the group's bandwidth is distributed evenly between the number of VFs in the group. If there are leftovers, they will be assigned to VFs whose individual rate limit has not been met yet.

VFs Rate Limit Feature Configuration

1. When VF rate group is supported by FW, the driver will create a new hierarchy in the SRI-OV sysfs named "groups" (`/sys/class/net/<ifname>/device/sriov/groups/`). It will contain all the info and the configurations allowed for VF groups.
2. All VFs are placed in group 0 by default since it is the only existing group following the initial driver start. It would be the only group available under `/sys/class/net/<ifname>/device/sriov/groups/`
3. The VF can be moved to a different group by writing to the group file -> `echo $GROUP_ID > /sys/class/net/<ifname>/device/sriov/<vf_id>/group`
4. The group IDs allowed are 0-255
5. Only when there is at least 1 VF in a group, there will be a group configuration available under `/sys/class/net/<ifname>/device/sriov/groups/` (Except for group 0, which is always available even when it's empty).
6. Once the group is created (by moving at least 1 VF to that group), users can configure the group's rate limit. For example:
 - a. `echo 10000 > /sys/class/net/<ifname>/device/sriov/5/max_tx_rate` - setting individual rate limitation of VF 5 to 10G (Optional)
 - b. `echo 7 > /sys/class/net/<ifname>/device/sriov/5/group` - moving VF 5 to group 7
 - c. `echo 5000 > /sys/class/net/<ifname>/device/sriov/groups/7/max_tx_rate` - setting group 7 with rate limitation of 5G
 - d. When running traffic via VF 5 now, it will be limited to 5G because of the group rate limit even though the VF itself is limited to 10G
 - e. `echo 3 > /sys/class/net/<ifname>/device/sriov/5/group` - moving VF 5 to group 3
 - f. Group 7 will now disappear from `/sys/class/net/<ifname>/device/sriov/groups` since there are 0 VFs in it. Group 3 will now appear. Since there's no rate limit on group 3, VF 5 can transmit at 10G (thanks to its individual configuration)

Notes:

- You can see to which group the VF belongs to in the 'stats' sysfs (`cat /sys/class/net/<ifname>/device/sriov/<vf_num>/stats`)
- You can see the current rate limit and number of attached VFs to a group in the group's 'config' sysfs (`cat /sys/class/net/<ifname>/device/sriov/groups/<group_id>/config`)

Bandwidth Guarantee per Group of VFs

Bandwidth guarantee (minimum BW) can be set on a group of VFs to ensure this group is able to transmit at least the amount of bandwidth specified on the wire.

Note the following:

- The minimum BW settings on VF groups determine how the groups share the total BW between themselves. It does not impact an individual VF's rate settings.
- The total minimum BW that is set on the VF groups should not exceed the total line rate. Otherwise, results are unexpected.
- It is still possible to set minimum BW on the individual VFs inside the group. This will determine how the VFs share the group's minimum BW between themselves. The total minimum BW of the VF member should not exceed the minimum BW of the group.

For instruction on how to create groups of VFs, see [Limit Bandwidth per Group of VFs](#) above.

Example

With a 40Gb link speed, assuming 4 groups and default group 0 have been created:

```
echo 20000 > /sys/class/net/<ifname>/device/sriov/group/1/min_tx_rate
echo 5000 > /sys/class/net/<ifname>/device/sriov/group/2/min_tx_rate
echo 15000 > /sys/class/net/<ifname>/device/sriov/group/3/min_tx_rate
```

```
Group 0(default) : 0 - No BW guarantee is configured.
Group 1 : 20000 - This is the maximum min rate among groups
Group 2 : 5000 which is 25% of the maximum min rate
Group 3 : 15000 which is 75% of the maximum min rate
Group 4 : 0 - No BW guarantee is configured.
```

Assuming there are VFs attempting to transmit in full line rate in all groups, the results would look like: In which case, the minimum BW allocation would be:

```
Group0 - Will have no BW to use since no BW guarantee was set on it while other groups do have such settings.
Group1 - Will transmit at 20Gb/s
Group2 - Will transmit at 5Gb/s
Group3 - Will transmit at 15Gb/s
Group4 - Will have no BW to use since no BW guarantee was set on it while other groups do have such settings.
```

Privileged VFs

In case a malicious driver is running over one of the VFs, and in case that VF's permissions are not restricted, this may open security holes. However, VFs can be marked as trusted and can thus receive an exclusive subset of physical function privileges or permissions. For example, in case of allowing all VFs, rather than specific VFs, to enter a promiscuous mode as a privilege, this will enable malicious users to sniff and monitor the entire physical port for incoming traffic, including traffic targeting other VFs, which is considered a severe security hole.

Privileged VFs Configuration

In the configuration example below, the VM is located on VF-0 and has the following MAC address: 11:22:33:44:55:66.

There are two ways to enable or disable trust:

1. Use the standard IP link commands - available from Kernel 4.5 and above.
 - a. To enable trust for a specific VF, run:

```
ip link set ens785f1 vf 0 trust on
```

- b. To disable trust for a specific VF, run:

```
ip link set ens785f1 vf 0 trust off
```

2. Specify echo "ON" or "OFF" to the file located under `/sys/class/net/<ETH_IF_NAME>/device/sriov/<VF index>/trust`.

- a. To enable trust for a specific VF, run:

```
echo "ON" > /sys/class/net/ens785f1/device/sriov/0/trust
```

- b. To disable trust for a specific VF, run:

```
echo "OFF" > /sys/class/net/ens785f1/device/sriov/0/trust
```

Probed VFs

Probing Virtual Functions (VFs) after SR-IOV is enabled might consume the adapter cards' resources. Therefore, it is recommended not to enable probing of VFs when no monitoring of the VM is needed. VF probing can be disabled in two ways, depending on the kernel version installed on your server:

1. If the kernel version installed is v4.12 or above, it is recommended to use the PCI sysfs interface `sriov_drivers_autoprobe`. For more information, see [linux-next branch](#).
2. If the kernel version installed is older than v4.12, it is recommended to use the `mlx5_core` module parameter `probe_vf` with driver version 4.1 or above.

Example:

```
echo 0 > /sys/module/mlx5_core/parameters/probe_vf
```

For more information on how to probe VFs, see [HowTo Configure and Probe VFs on mlx5 Drivers](#) Community post.

2.3.2.1.4.6 VF Promiscuous Rx Modes

VF Promiscuous Mode

VFs can enter a promiscuous mode that enables receiving the unmatched traffic and all the multicast traffic that reaches the physical port in addition to the traffic originally targeted to the VF. The unmatched traffic is any traffic's DMAC that does not match any of the VFs' or PFs' MAC addresses.

Note: Only privileged/trusted VFs can enter the VF promiscuous mode.

➤ To set the promiscuous mode on for a VF, run:

```
ifconfig eth2 promisc
```

➤ To exit the promiscuous mode, run:

```
ifconfig eth2 -promisc
```

VF All-Multi Mode

VFs can enter an all-multi mode that enables receiving all the multicast traffic sent from/to the other functions on the same physical port in addition to the traffic originally targeted to the VF. Note: Only privileged/trusted VFs can enter the all-multi RX mode.

➤ To set the all-multi mode on for a VF, run:

```
ifconfig eth2 allmulti
```

➤ To exit the all-multi mode, run:

```
#ifconfig eth2 -allmulti
```

2.3.2.1.5 Uninstalling the SR-IOV Driver

➤ To uninstall SR-IOV driver, perform the following:

1. For Hypervisors, detach all the Virtual Functions (VF) from all the Virtual Machines (VM) or stop the Virtual Machines that use the Virtual Functions. Please be aware that stopping the driver when there are VMs that use the VFs, will cause machine to hang.
2. Run the script below. Please be aware, uninstalling the driver deletes the entire driver's file, but does not unload the driver.

```
[root@swl022 ~]# /usr/sbin/ofed_uninstall.sh
This program will uninstall all OFED packages on your machine.
Do you want to continue?[y/N]:y
Running /usr/sbin/vendor_pre_uninstall.sh
Removing OFED Software installations
Running /bin/rpm -e --allmatches kernel-ib kernel-ib-devel libibverbs libibverbs-devel libibverbs-
devel-static libibverbs-utils libmlx4 libmlx4-devel libibcm libibcm-devel libibumad libibumad-devel
libibumad-static libibmad libibmad-devel libibmad-static librdmacm librdmacm-utils librdmacm-devel ibacm
opensm-libs opensm-devel perftest compat-dapl compat-dapl-devel dapl dapl-devel dapl-devel-static dapl-
utils srptools infiniband-diags-guest ofed-scripts opensm-devel
warning: /etc/infiniband/openib.conf saved as /etc/infiniband/openib.conf.rpmsave
Running /tmp/2818-ofed_vendor_post_uninstall.sh
```

3. Restart the server.

2.3.2.2 Enabling Paravirtualization

➤ To enable Paravirtualization:

The example below works on RHEL7.* without a Network Manager.

1. Create a bridge.

```
vim /etc/sysconfig/network-scripts/ifcfg-bridge0
DEVICE=bridge0
TYPE=Bridge
```

```
IPADDR=12.195.15.1
NETMASK=255.255.0.0
BOOTPROTO=static
ONBOOT=yes
NM_CONTROLLED=no
DELAY=0
```

2. Change the related interface (in the example below bridge0 is created over eth5).

```
DEVICE=eth5
BOOTPROTO=none
STARTMODE=on
HWADDR=00:02:c9:2e:66:52
TYPE=Ethernet
NM_CONTROLLED=no
ONBOOT=yes
BRIDGE=bridge0
```

3. Restart the service network.
4. Attach a bridge to VM.

```
ifconfig -a
...
eth6      Link encap:Ethernet  HWaddr 52:54:00:E7:77:99
          inet addr:13.195.15.5  Bcast:13.195.255.255  Mask:255.255.0.0
          inet6 addr: fe80::5054:ff:fee7:7799/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:481 errors:0 dropped:0 overruns:0 frame:0
          TX packets:450 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:22440 (21.9 KiB)  TX bytes:19232 (18.7 KiB)
          Interrupt:10 Base address:0xa000
...
```

2.3.2.3 VXLAN Hardware Stateless Offloads

VXLAN technology provides scalability and security challenges solutions. It requires extension of the traditional stateless offloads to avoid performance drop. ConnectX family cards offer the following stateless offloads for a VXLAN packet, similar to the ones offered to non-encapsulated packets. VXLAN protocol encapsulates its packets using outer UDP header.


Available hardware stateless offloads:

- Checksum generation (Inner IP and Inner TCP/UDP)
- Checksum validation (Inner IP and Inner TCP/UDP)
- TSO support for inner TCP packets
- RSS distribution according to inner packets attributes
- Receive queue selection - inner frames may be steered to specific QPs

2.3.2.3.1

Enabling VXLAN Hardware Stateless Offloads

VXLAN offload is enabled by default for ConnectX-4 family devices running the minimum required firmware version and a kernel version that includes VXLAN support.

 To confirm if the current setup supports VXLAN, run:

```
ethtool -k $DEV | grep udp_tnl
```

Example:

```
ethtool -k ens1f0 | grep udp_tnl
tx-udp_tnl-segmentation: on
```


ConnectX-4 family devices support configuring multiple UDP ports for VXLAN offload. Ports can be added to the device by configuring a VXLAN device from the OS command line using the "ip" command.

Note: If you configure multiple UDP ports for offload and exceed the total number of ports supported by hardware, then those additional ports will still function properly, but will not benefit from any of the stateless offloads.

Example:

```
ip link add vxlan0 type vxlan id 10 group 239.0.0.10 ttl 10 dev ens1f0 dstport 4789
ip addr add 192.168.4.7/24 dev vxlan0
ip link set up vxlan0
```

Note: 'dstport' parameters are not supported in Ubuntu 14.4.

The VXLAN ports can be removed by deleting the VXLAN interfaces.

Example:

```
ip link delete vxlan0
```

2.3.2.3.2 Important Note

VXLAN tunneling adds 50 bytes (14-eth + 20-ip + 8-udp + 8-vxlan) to the VM Ethernet frame. Please verify that either the MTU of the NIC who sends the packets, e.g. the VM virtio-net NIC or the host side veth device or the uplink takes into account the tunneling overhead. Meaning, the MTU of the sending NIC has to be decremented by 50 bytes (e.g 1450 instead of 1500), or the uplink NIC MTU has to be incremented by 50 bytes (e.g 1550 instead of 1500)

2.3.2.4 Q-in-Q Encapsulation per VF in Linux (VST)

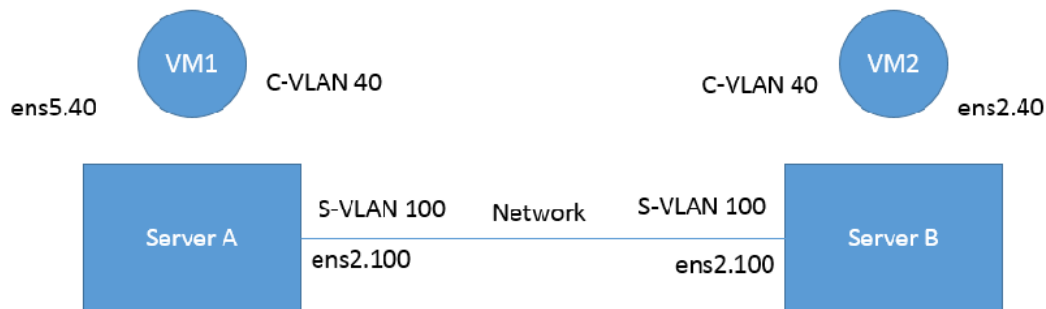
This feature is supported on ConnectX-5 and ConnectX-6 adapter cards only.

ConnectX-4 and ConnectX-4 Lx adapter cards support 802.1Q double-tagging (C-tag stacking on C-tag) - refer to "[802.1Q Double-Tagging](#)" section.

This section describes the configuration of IEEE 802.1ad QinQ VLAN tag (S-VLAN) to the hypervisor per Virtual Function (VF). The Virtual Machine (VM) attached to the VF (via SR-IOV) can send traffic with or without C-VLAN. Once a VF is configured to VST QinQ encapsulation (VST QinQ), the adapter's hardware will insert S-VLAN to any packet from the VF to the physical port. On the receive side, the adapter hardware will strip the S-VLAN from any packet coming from the wire to that VF.

2.3.2.4.1 Setup

The setup assumes there are two servers equipped with ConnectX-5/ConnectX-6 adapter cards.



2.3.2.4.2 Prerequisites

- Kernel must be of v3.10 or higher, or custom/inbox kernel must support vlan-stag
- Firmware version 16/20.21.0458 or higher must be installed for ConnectX-5/ConnectX-6 HCAs
- The server should be enabled in SR-IOV and the VF should be attached to a VM on the hypervisor.
 - In order to configure SR-IOV in Ethernet mode for ConnectX-5/ConnectX-6 adapter cards, please refer to "[Configuring SR-IOV for ConnectX-4/ConnectX-5 \(Ethernet\)](#)" section. In the following configuration example, the VM is attached to VF0.
- Network Considerations - the network switches may require increasing the MTU (to support 1522 MTU size) on the relevant switch ports.

2.3.2.4.3 Configuring Q-in-Q Encapsulation per Virtual Function for ConnectX-5/ConnectX-6

1. Add the required S-VLAN (QinQ) tag (on the hypervisor) per port per VF. There are two ways to add the S-VLAN:
 - a. By using sysfs:

```
echo '100:0:802.1ad' > /sys/class/net/ens1f0/device/sriov/0/vlan
```

- b. By using the ip link command (available only when using the latest Kernel version):

```
ip link set dev ens1f0 vf 0 vlan 100 proto 802.1ad
```

Check the configuration using the ip link show command:

```
# ip link show ens1f0
ens1f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT qlen 1000
    link/ether ec:0d:9a:44:37:84 brd ff:ff:ff:ff:ff:ff
    vf 0 MAC 00:00:00:00:00:00, vlan 100, vlan protocol 802.1ad, spoof checking off, link-state
    auto, trust off
    vf 1 MAC 00:00:00:00:00:00, spoof checking off, link-state auto, trust off
    vf 2 MAC 00:00:00:00:00:00, spoof checking off, link-state auto, trust off
    vf 3 MAC 00:00:00:00:00:00, spoof checking off, link-state auto, trust off
    vf 4 MAC 00:00:00:00:00:00, spoof checking off, link-state auto, trust off
```

2. **Optional:** Add S-VLAN priority. Use the qos parameter in the ip link command (or sysfs):

```
ip link set dev ens1f0 vf 0 vlan 100 qos 3 proto 802.1ad
```

Check the configuration using the ip link show command:

```
# ip link show ens1f0
ens1f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT qlen 1000
link/ether ec:0d:9a:44:37:84 brd ff:ff:ff:ff:ff:ff
vf 0 MAC 00:00:00:00:00:00, vlan 100, qos 3, vlan protocol 802.1ad, spoof checking off, link-state
auto, trust off
vf 1 MAC 00:00:00:00:00:00, spoof checking off, link-state auto, trust off
vf 2 MAC 00:00:00:00:00:00, spoof checking off, link-state auto, trust off
vf 3 MAC 00:00:00:00:00:00, spoof checking off, link-state auto, trust off
vf 4 MAC 00:00:00:00:00:00, spoof checking off, link-state auto, trust off
```

3. Create a VLAN interface on the VM and add an IP address.

```
ip link add link ens5 ens5.40 type vlan protocol 802.1q id 40
ip addr add 42.134.135.7/16 brd 42.134.255.255 dev ens5.40
ip link set dev ens5.40 up
```

4. To verify the setup, run ping between the two VMs and open Wireshark or tcpdump to capture the packet.

2.3.2.5 802.1Q Double-Tagging

This section describes the configuration of 802.1Q double-tagging support to the hypervisor per Virtual Function (VF). The Virtual Machine (VM) attached to the VF (via SR-IOV) can send traffic with or without C-VLAN. Once a VF is configured to VST encapsulation, the adapter's hardware will insert C-VLAN to any packet from the VF to the physical port. On the receive side, the adapter hardware will strip the C-VLAN from any packet coming from the wire to that VF.

2.3.2.5.1 Configuring 802.1Q Double-Tagging per Virtual Function

1. Add the required C-VLAN tag (on the hypervisor) per port per VF. There are two ways to add the C-VLAN:
 - a. By using sysfs:

```
echo '100:0:802.1q' > /sys/class/net/ens1f0/device/sriov/0/vlan
```

- b. By using the ip link command (available only when using the latest Kernel version):

```
ip link set dev ens1f0 vf 0 vlan 100
```

Check the configuration using the ip link show command:

```
# ip link show ens1f0
ens1f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT qlen 1000
link/ether ec:0d:9a:44:37:84 brd ff:ff:ff:ff:ff:ff
vf 0 MAC 00:00:00:00:00:00, vlan 100, spoof checking off, link-state auto, trust off
vf 1 MAC 00:00:00:00:00:00, spoof checking off, link-state auto, trust off
vf 2 MAC 00:00:00:00:00:00, spoof checking off, link-state auto, trust off
vf 3 MAC 00:00:00:00:00:00, spoof checking off, link-state auto, trust off
vf 4 MAC 00:00:00:00:00:00, spoof checking off, link-state auto, trust off
```

2. Create a VLAN interface on the VM and add an IP address.

```
# ip link add link ens5 ens5.40 type vlan protocol 802.1q id 40
# ip addr add 42.134.135.7/16 brd 42.134.255.255 dev ens5.40
# ip link set dev ens5.40 up
```

3. To verify the setup, run ping between the two VMs and open Wireshark or tcpdump to capture the packet.

2.3.3 Resiliency

The chapter contains the following sections:

- [Reset Flow](#)

2.3.3.1 Reset Flow

Reset Flow is activated by default. Once a "fatal device" error is recognized, both the HCA and the software are reset, the ULPs and user application are notified about it, and a recovery process is performed once the event is raised.

Currently, a reset flow can be triggered by a firmware assert with Recover Flow Request (RFR) only. Firmware RFR support should be enabled explicitly using `mlxconfig` commands.

➤ *To query the current value, run:*

```
mlxconfig -d /dev/mst/mt4115_pciconf0 query | grep SW_RECOVERY_ON_ERRORS
```

➤ *To enable RFR bit support, run:*

```
mlxconfig -d /dev/mst/mt4115_pciconf0 set SW_RECOVERY_ON_ERRORS=true
```

2.3.3.1.1 Kernel ULPs

Once a "fatal device" error is recognized, an `IB_EVENT_DEVICE_FATAL` event is created, ULPs are notified about the incident, and outstanding WQEs are simulated to be returned with "flush in error" message to enable each ULP to close its resources and not get stuck via calling its "remove_one" callback as part of "Reset Flow".

Once the unload part is terminated, each ULP is called with its " `add_one` " callback, its resources are re-initialized and it is re-activated.

2.3.3.1.2 SR-IOV

If the Physical Function recognizes the error, it notifies all the VFs about it by marking their communication channel with that information, consequently, all the VFs and the PF are reset. If the VF encounters an error, only that VF is reset, whereas the PF and other VFs continue to work unaffected.

2.3.3.1.3 Forcing the VF to Reset

If an outside "reset" is forced by using the PCI sysfs entry for a VF, a reset is executed on that VF once it runs any command over its communication channel.

For example, the below command can be used on a hypervisor to reset a VF defined by 0000:04:00.1:

```
echo 1 >/sys/bus/pci/devices/0000:04:00.1/reset
```

2.3.3.1.4 Extended Error Handling (EEH)

Extended Error Handling (EEH) is a PowerPC mechanism that encapsulates AER, thus exposing AER events to the operating system as EEH events.

The behavior of ULPs and user space applications is identical to the behavior of AER.

2.3.3.1.5 CRDUMP

CRDUMP feature allows for taking an automatic snapshot of the device CR-Space in case the device's FW/HW fails to function properly.

Snapshots Triggers:

The snapshot is triggered after firmware detects a critical issue, requiring a recovery flow.

This snapshot can later be investigated and analyzed to track the root cause of the failure. Currently, only the first snapshot is stored, and is exposed using a temporary virtual file. The virtual file is cleared upon driver reset.

When a critical event is detected, a message indicating CRDUMP collection will be printed to the Linux log. User should then back up the file pointed to in the printed message. The file location format is: `/proc/driver/mlx5_core/crdump/<pci address>`

Snapshot should be copied by Linux standard tool for future investigation.

2.3.3.1.6 Firmware Tracer

This mechanism allows for the device's FW/HW to log important events into the event tracing system (`/sys/kernel/debug/tracing`) without requiring any NVIDIA tool.

To be able to use this feature, trace points must be enabled in the kernel.

This feature is enabled by default, and can be controlled using sysfs commands.

➤ *To disable the feature:*

```
echo 0 > /sys/kernel/debug/tracing/events/mlx5/fw_tracer/enable
```

➤ *To enable the feature:*

```
echo 1 > /sys/kernel/debug/tracing/events/mlx5/fw_tracer/enable
```

➤ To view FW traces using vim text editor:

```
vim /sys/kernel/debug/tracing/trace
```

2.3.4 Docker Containers

On Linux, Docker uses resource isolation of the Linux kernel, to allow independent "containers" to run within a single Linux kernel instance.

Docker containers are supported on MLNX-EN using Docker runtime. Virtual RoCE and InfiniBand devices are supported using SR-IOV mode.

Currently, RDMA/RoCE devices are supported in the modes listed in the following table.

Linux Containers Networking Modes

Orchestration and Clustering Tool	Version	Networking Mode	Link Layer	Virtualization Mode
Docker	Docker Engine 17.03 or higher	SR-IOV using sriov-plugin along with docker run wrapper tool	InfiniBand and Ethernet	SR-IOV
Kubernetes	Kubernetes 1.10.3 or higher	SR-IOV using device plugin, and using SR-IOV CNI plugin	InfiniBand and Ethernet	SR-IOV
		VXLAN using IPoIB bridge	InfiniBand	Shared HCA

2.3.4.1 Docker Using SR-IOV

In this mode, Docker engine is used to run containers along with SR-IOV networking plugin. To isolate the virtual devices, `docker_rdma_sriov` tool should be used. This mode is applicable to both InfiniBand and Ethernet link layers.

To obtain the plugin, visit: hub.docker.com/r/rdma/sriov-plugin

To install the `docker_rdma_sriov` tool, use the container tools installer available via hub.docker.com/r/rdma/container_tools_installer

For instructions on how to use Docker with SR-IOV, refer to [Docker RDMA SRIOV Networking with ConnectX4/ConnectX5/ConnectX6](#) Community post.

2.3.4.2 Kubernetes Using SR-IOV

In order to use RDMA in Kubernetes environment with SR-IOV networking mode, two main components are required:

1. RDMA device plugin - this plugin allows for exposing RDMA devices in a Pod
2. SR-IOV CNI plugin - this plugin provisions VF net device in a Pod

When used in SR-IOV mode, this plugin enables SR-IOV and performs necessary configuration including setting GUID, MAC, privilege mode, and Trust mode.

The plugin also allocates the VF devices when Pods are scheduled and requested by Kubernetes

framework.

2.3.4.3 Kubernetes with Shared HCA

One RDMA device (HCA) can be shared among multiple Pods running in a Kubernetes worker nodes. User defined networks are created using VXLAN or VETH networking devices. RDMA device (HCA) can be shared among multiple Pods running in a Kubernetes worker nodes.

2.3.5 Fast Driver Unload

This feature enables optimizing mlx5 driver teardown time in shutdown and kexec flows.

The fast driver unload is disabled by default. To enable it, the `prof_sel` module parameter of `mlx5_core` module should be set to 3.

2.3.6 OVS Offload Using ASAP² Direct

2.3.6.1 Overview

Supported on ConnectX-5 and above adapter cards.

Open vSwitch (OVS) allows Virtual Machines (VMs) to communicate with each other and with the outside world. OVS traditionally resides in the hypervisor and switching is based on twelve tuple matching on flows. The OVS software based solution is CPU intensive, affecting system performance and preventing full utilization of the available bandwidth.

NVIDIA Accelerated Switching And Packet Processing (ASAP²) technology allows OVS offloading by handling OVS data-plane in ConnectX-5 onwards NIC hardware (Embedded Switch or eSwitch) while maintaining OVS control-plane unmodified. As a result, we observe significantly higher OVS performance without the associated CPU load.

As of v5.0, OVS-DPDK became part of `MLNX_EN` package. OVS-DPDK supports ASAP² just as the OVS-Kernel (Traffic Control (TC) kernel-based solution) does, yet with a different set of features.

The traditional ASAP² hardware data plane is built over SR-IOV virtual functions (VFs), so that the VF is passed through directly to the VM, with the NVIDIA driver running within the VM. An alternate approach that is also supported is vDPA (vhost Data Path Acceleration). vDPA allows the connection to the VM to be established using VirtIO, so that the data-plane is built between the SR-IOV VF and the standard VirtIO driver within the VM, while the control-plane is managed on the host by the vDPA application. Two flavors of vDPA are supported, Software vDPA; and Hardware vDPA. Software vDPA management functionality is embedded into OVS-DPDK, while Hardware vDPA uses a standalone application for management, and can be run with both OVS-Kernel and OVS-DPDK. For further information, please see sections [VirtIO Acceleration through VF Relay \(Software vDPA\)](#) and [VirtIO Acceleration through Hardware vDPA](#).

2.3.6.2 Installing OVS-Kernel ASAP² Packages

Install the required packages. For the complete solution, you need to install supporting `MLNX_EN`(v4.4 and above), `iproute2`, and `openvswitch` packages.

2.3.6.3 Installing OVS-DPDK ASAP² Packages

Run:

```
./install --ovs-dpdk -upstream-libs
```

2.3.6.4 Setting Up SR-IOV

Note that this section applies to both OVS-DPDK and OVS-Kernel similarly.

➤ *To set up SR-IOV:*

1. Choose the desired card.

The example below shows a dual-ported ConnectX-5 card (device ID 0x1017) and a single SR-IOV VF (Virtual Function, device ID 0x1018).

In SR-IOV terms, the card itself is referred to as the PF (Physical Function).

```
# lspci -nn | grep Mellanox
0a:00.0 Ethernet controller [0200]: Mellanox Technologies MT27800 Family [ConnectX-5] [15b3:1017]
0a:00.1 Ethernet controller [0200]: Mellanox Technologies MT27800 Family [ConnectX-5] [15b3:1017]
0a:00.2 Ethernet controller [0200]: Mellanox Technologies MT27800 Family [ConnectX-5 Virtual Function]
[15b3:1018]
```

Enabling SR-IOV and creating VFs is done by the firmware upon admin directive as explained in Step 5 below.

2. Identify the NVIDIA NICs and locate net-devices which are on the NIC PCI BDF.

```
# ls -l /sys/class/net/ | grep 04:00
lrwxrwxrwx 1 root root 0 Mar 27 16:58 enp4s0f0 -> ../../devices/pci0000:00/0000:00:03.0/0000:04:00.0/net/enp4s0f0
lrwxrwxrwx 1 root root 0 Mar 27 16:58 enp4s0f1 -> ../../devices/pci0000:00/0000:00:03.0/0000:04:00.1/net/enp4s0f1
lrwxrwxrwx 1 root root 0 Mar 27 16:58 eth0 -> ../../devices/pci0000:00/0000:00:03.0/0000:04:00.2/net/eth0
lrwxrwxrwx 1 root root 0 Mar 27 16:58 eth1 -> ../../devices/pci0000:00/0000:00:03.0/0000:04:00.3/net/eth1
```

The PF NIC for port #1 is enp4s0f0, and the rest of the commands will be issued on it.

3. Check the firmware version.

Make sure the firmware versions installed are as state in the Release Notes document.

```
# ethtool -i enp4s0f0 | head -5
driver: mlx5_core
version: 5.0-5
firmware-version: 16.21.0338
expansion-rom-version:
bus-info: 0000:04:00.0
```

4. Make sure SR-IOV is enabled on the system (server, card).

Make sure SR-IOV is enabled by the server BIOS, and by the firmware with up to N VFs, where N is the number of VFs required for your environment. Refer to "[NVIDIA Firmware Tools](#)" below for more details.


```
# cat /sys/class/net/enp4s0f0/device/sriov_totalvfs
4
```

5. Turn ON SR-IOV on the PF device.

```
# echo 2 > /sys/class/net/enp4s0f0/device/sriov_numvfs
```

6. Provision the VF MAC addresses using the IP tool.

```
# ip link set enp4s0f0 vf 0 mac e4:11:22:33:44:50
# ip link set enp4s0f0 vf 1 mac e4:11:22:33:44:51
```

7. Verify the VF MAC addresses were provisioned correctly and SR-IOV was turned ON.

```
# cat /sys/class/net/enp4s0f0/device/sriov_numvfs
2

# ip link show dev enp4s0f0
256: enp4s0f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq master ovs-system state UP mode DEFAULT
group default qlen 1000
    link/ether e4:1d:2d:60:95:a0 brd ff:ff:ff:ff:ff:ff
    vf 0 MAC e4:11:22:33:44:50, spoof checking off, link-state auto
    vf 1 MAC e4:11:22:33:44:51, spoof checking off, link-state auto
```

In the example above, the maximum number of possible VFs supported by the firmware is 4 and only 2 are enabled.

8. Provision the PCI VF devices to VMs using PCI Pass-Through or any other preferred virt tool of choice, e.g virt-manager.

For further information on SR-IOV, refer to [HowTo Configure SR-IOV for ConnectX-4/ConnectX-5/ConnectX-6 with KVM \(Ethernet\)](#).

2.3.6.5 OVS Hardware Offloads Configuration

2.3.6.5.1 OVS-Kernel Hardware Offloads

2.3.6.5.1.1 SwitchDev Configuration

1. Unbind the VFs.

```
echo 0000:04:00.2 > /sys/bus/pci/drivers/mlx5_core/unbind
echo 0000:04:00.3 > /sys/bus/pci/drivers/mlx5_core/unbind
```

VMs with attached VFs must be powered off to be able to unbind the VFs.

2. Change the eSwitch mode from Legacy to SwitchDev on the PF device.
This will also create the VF representor netdevices in the host OS.

```
# devlink dev eswitch set pci/0000:3b:00.0 mode switchdev
```

Before changing the mode, make sure that all VFs are unbound.

To go back to SR-IOV legacy mode, run:

```
# devlink dev eswitch set pci/0000:3b:00.0 mode legacy
```

This will also remove the VF representor netdevices.

On old OSs or kernels that do not support Devlink, moving to SwitchDev mode can be done using sysfs.

```
# echo switchdev > /sys/class/net/enp4s0f0/compat/devlink/mode
```

3. At this stage, VF representors have been created. To map representor to its VF, make sure to obtain the representor's switchid and portname from:

```
# ip -d link show eth4
41: enp0s8f0_1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT group default
qlen 1000
    link/ether ba:e6:21:37:bc:d4 brd ff:ff:ff:ff:ff:ff promiscuity 0 addrngenmode eui64 numtxqueues 10
    numrxqueues 10 gso_max_size 65536 gso_max_segs 65535 portname pf0vf1 switchid f4ab580003a1420c
```

switchid - used to map representor to device, both device PFs have the same switchid.

portname - used to map representor to PF and VF, value returned is pfXvfY, where X is the PF number and Y is the number of VF.

On old kernels, switchid and portname can be acquired through sysfs:

4. Bind the VFs.

```
echo 0000:04:00.2 > /sys/bus/pci/drivers/mlx5_core/bind
echo 0000:04:00.3 > /sys/bus/pci/drivers/mlx5_core/bind
```

2.3.6.5.1.2 SwitchDev Performance Tuning

SwitchDev performance can be further improved by tuning it.

Steering Mode

OVS-kernel supports two steering modes for rules insertion into hardware.

1. SMFS - Software Managed Flow Steering (as of MLNX_OFED v5.1, this is the default mode)
Rules are inserted directly to the hardware by the software (driver). This mode is optimized for rules insertion.
2. DMFS - Device Managed Flow Steering
Rules insertion is done using firmware commands. This mode is optimized for throughput with a small amount of rules in the system.
The mode can be controlled via sysfs or devlink API in kernels that support it:

```
Sysfs:
# echo smfs > /sys/class/net/<PF netdev>/compat/devlink/steering_mode
Devlink:
```

```
# devlink dev param set pci/0000:00:08.0 name flow_steering_mode value "smfs" cmode runtime
Replace smfs param with dmfs for device managed flow steering
```

Notes:

- The mode should be set prior to moving to SwitchDev, by echoing to the sysfs or invoking the devlink command.
- Only when moving to SwitchDev will the driver use the mode set by the previous step.
- Mode cannot be changed after moving to SwitchDev.
- The steering mode is applicable for SwitchDev mode only, meaning it does not affect legacy SR-IOV or other configurations.

Troubleshooting SMFS

mlx5 debugfs was extended to support presenting Software Steering resources: dr_domain including it's tables, matchers and rules. The interface is read-only.

While dump is being created, new steering rules cannot be inserted/deleted. The steering information is dumped in the CSV form with the following format:

```
<object_type>,<object_ID>, <object_info>,...,<object_info>
```

This data can be read at the following path: `/sys/kernel/debug/mlx5/<BDF>/steering/fdb/<domain_handle>`

Example:

```
# cat /sys/kernel/debug/mlx5/0000:82:00.0/steering/fdb/dmn_000018644
3100,0x55caa4621c50,0xee802,4,65533
3101,0x55caa4621c50,0xe0100008
```

You can then use the steering dump parser to make the output more human readable. The parser can be found in the following public GitHub repository: https://github.com/Mellanox/mlx_steering_dump

vPort Match Mode

OVS-kernel support two modes that define how the rules on match on vport.

1. Metadata - rules match on metadata instead of vport number (default mode). This mode is needed in order to support SR-IOV Live migration and Dual port RoCE features. Matching on Metadata can have a performance impact.
2. Legacy - rules match on vport number. In this mode, performance can be higher in comparison to Metadata. It can still be used only if none of the above features (SR-IOV Live migration and Dual port RoCE) is enabled/used. The mode can be controlled via sysfs:

```
Set Legacy:
# echo legacy > /sys/class/net/<PF netdev>/compat/devlink/vport_match_mode

Set metadata:
Devlink:
# echo metadata > /sys/class/net/<PF netdev>/compat/devlink/vport_match_mode
```

Note: This mode should be set prior to moving to SwitchDev, by echoing to the sysfs.

Flow Table Large Group Number

Offloaded flows, including Connection Tracking, are added to Virtual Switch Forwarding Data Base (FDB) flow tables. FDB tables have a set of flow groups, where each flow group saves the same traffic pattern flows. E.g, for connection tracking offloaded flow, TCP and UDP are different traffic patterns which will end up in two different flow groups.

A flow group has a limited size to save flow entries. As default, the driver has 15 big FDB flow groups. Each of these big flow groups can save $4M / (15 + 1) = 256k$ different 5-tuple flow entries at most. For scenarios with more than 15 traffic patterns, the driver provides a module parameter (`num_of_groups`) to allow customization and performance tuning.

The mode can be controlled via module param or devlink API for kernels that support it:

```
Module param:
# echo <num_of_groups> > /sys/module/mlx5_core/parameters/num_of_groups

Devlink:
# devlink dev param set pci/0000:82:00.0 name fdb_large_groups \
  cmode driverinit value 20
```

Notes:

- In MLNX_OFED v5.1, the default value was changed from 4 to 15.
- The change takes effect immediately if there is no flow inside the FDB table (no traffic running and all offloaded flows are aged out). And it can be dynamically changed without reloading the driver.
If there are still offloaded flows residual when changing this parameter, it will only take effect after all flows have aged out.

2.3.6.5.1.3 Open vSwitch Configuration

Open vSwitch configuration is a simple OVS bridge configuration with SwitchDev.

1. Run the openvswitch service.

```
# systemctl start openvswitch
```

2. Create an OVS bridge (here it's named ovs-sriov).

```
# ovs-vsctl add-br ovs-sriov
```

3. Enable hardware offload (disabled by default).

```
# ovs-vsctl set Open_vSwitch . other_config:hw-offload=true
```

4. Restart the openvswitch service. This step is required for HW offload changes to take effect.

```
# systemctl restart openvswitch
```

HW offload policy can also be changed by setting the tc-policy using one on the following values:

- * none - adds a TC rule to both the software and the hardware (default)

```
* skip_sw - adds a TC rule only to the hardware
* skip_hw - adds a TC rule only to the software
The above change is used for debug purposes.
```

5. Add the PF and the VF representor netdevices as OVS ports.

```
# ovs-vsctl add-port ovs-sriov enp4s0f0
# ovs-vsctl add-port ovs-sriov enp4s0f0_0
# ovs-vsctl add-port ovs-sriov enp4s0f0_1
```

Make sure to bring up the PF and representor netdevices.

```
# ip link set dev enp4s0f0 up
# ip link set dev enp4s0f0_0 up
# ip link set dev enp4s0f0_1 up
```

The PF represents the uplink (wire).

```
# ovs-dpctl show
system@ovs-system:
  lookups: hit:0 missed:192 lost:1
  flows: 2
  masks: hit:384 total:2 hit/pkt:2.00
  port 0: ovs-system (internal)
  port 1: ovs-sriov (internal)
  port 2: enp4s0f0
  port 3: enp4s0f0_0
  port 4: enp4s0f0_1
```

6. Run traffic from the VFs and observe the rules added to the OVS data-path.

```
# ovs-dpctl dump-flows

recirc_id(0),in_port(3),eth(src=e4:11:22:33:44:50,dst=e4:1d:2d:a5:f3:9d),
eth_type(0x0800),ipv4(frag=no), packets:33, bytes:3234, used:1.196s, actions:2

recirc_id(0),in_port(2),eth(src=e4:1d:2d:a5:f3:9d,dst=e4:11:22:33:44:50),
eth_type(0x0800),ipv4(frag=no), packets:34, bytes:3332, used:1.196s, actions:3
```

In the example above, the ping was initiated from VF0 (OVS port 3) to the outer node (OVS port 2), where the VF MAC is e4:11:22:33:44:50 and the outer node MAC is e4:1d:2d:a5:f3:9d. As shown above, two OVS rules were added, one in each direction.

Note that you can also verify offloaded packets by adding type=offloaded to the command. For example:

```
# ovs-appctl dpctl/dump-flows type=offloaded
```

2.3.6.5.1.4 Open vSwitch Performance Tuning

Flow Aging

The aging timeout of OVS is given in ms and can be controlled using the following command.

```
# ovs-vsctl set Open_vSwitch . other_config:max-idle=30000
```

TC Policy

Specifies the policy used with HW offloading.

- none - adds a TC rule to both the software and the hardware (default)
- skip_sw - adds a TC rule only to the hardware

- skip_hw - adds a TC rule only to the software

Example:

```
# ovs-vsctl set Open_vSwitch . other_config:tc-policy=skip_sw
```

Note: TC policy should only be used for debugging purposes.

Max-Revalidator

Specifies the maximum time (in ms) that revalidator threads will wait for kernel statistics before executing flow revalidation.

```
# ovs-vsctl set Open_vSwitch . other_config:max-revalidator=10000
```

n-handler-threads

Specifies the number of threads for software datapaths to use for handling new flows. The default value is the number of online CPU cores minus the number of revalidators.

```
# ovs-vsctl set Open_vSwitch . other_config:n-handler-threads=4
```

n-revalidator-threads

Specifies the number of threads for software datapaths to use for revalidating flows in the datapath.

```
# ovs-vsctl set Open_vSwitch . other_config:n-revalidator-threads=4
```

vlan-limit

Limits the number of VLAN headers that can be matched to the specified number.

```
# ovs-vsctl set Open_vSwitch . other_config:vlan-limit=2
```

2.3.6.5.1.5 Basic TC Rules Configuration

Offloading rules can also be added directly, and not only through OVS, using the tc utility.



To create an offloading rule using TC:

1. Create an ingress qdisc (queueing discipline) for each interface that you wish to add rules into.

```
# tc qdisc add dev enp4s0f0 ingress
# tc qdisc add dev enp4s0f0_0 ingress
# tc qdisc add dev enp4s0f0_1 ingress
```

2. Add TC rules using flower classifier in the following format.

```
# tc filter add dev NETDEVICE ingress protocol PROTOCOL prio PRIORITY \
[chain CHAIN] flower [ MATCH_LIST ] [ action ACTION_SPEC ]
```

Note: List of supported matches (specifications) and actions can be found in [Classification Fields \(Matches\)](#) section.

3. Dump the existing tc rules using flower classifier in the following format.

```
# tc [ -s ] filter show dev NETDEVICE ingress
```

2.3.6.5.1.6 SR-IOV VF LAG

SR-IOV VF LAG allows the NIC's physical functions (PFs) to get the rules that the OVS will try to offload to the bond net-device, and to offload them to the hardware e-switch. Bond modes supported are:

- Active-Backup
- XOR
- LACP

SR-IOV VF LAG enables complete offload of the LAG functionality to the hardware. The bonding creates a single bonded PF port. Packets from up-link can arrive from any of the physical ports, and will be forwarded to the bond device.

When hardware offload is used, packets from both ports can be forwarded to any of the VFs. Traffic from the VF can be forwarded to both ports according to the bonding state. Meaning, when in active-backup mode, only one PF is up, and traffic from any VF will go through this PF. When in XOR or LACP mode, if both PFs are up, traffic from any VF will split between these two PFs.

SR-IOV VF LAG Configuration on ASAP²

To enable SR-IOV VF LAG, both physical functions of the NIC should first be configured to SR-IOV SwitchDev mode, and only afterwards bond the up-link representors.

The example below shows the creation of bond interface on two PFs:

1. Load bonding device and enslave the up-link representor (currently PF) net-device devices.

```
modprobe bonding mode=802.3ad
Ifup bond0 (make sure ifcfg file is present with desired bond configuration)
ip link set enp4s0f0 master bond0
ip link set enp4s0f1 master bond0
```

2. Add the VF representor net-devices as OVS ports. If tunneling is not used, add the bond device as well.

```
ovs-vsctl add-port ovs-sriov bond0
ovs-vsctl add-port ovs-sriov enp4s0f0_0
ovs-vsctl add-port ovs-sriov enp4s0f1_0
```

3. Make sure to bring up the PF and the representor netdevices.

```
ip link set dev bond0 up
ip link set dev enp4s0f0_0 up
ip link set dev enp4s0f1_0 up
```

Once SR-IOV VF LAG is configured, all VFs of the two PFs will become part of the bond, and will behave as described above.

Limitations

- In VF LAG mode, outgoing traffic in load balanced mode is according to the origin ring, thus, half of the rings will be coupled with port 1 and half with port 2. All the traffic on the same ring will be sent from the same port.
- VF LAG configuration is not supported when the NUM_OF_VFS configured in mlxconfig is higher than 64.

Using TC with VF LAG

Both rules can be added using either of the following.

1. Shared block (supported from kernel 4.16 and RHEL/CentOS 7.7 and above).

```
# tc qdisc add dev bond0 ingress_block 22 ingress
# tc qdisc add dev ens4p0 ingress_block 22 ingress
# tc qdisc add dev ens4p1 ingress_block 22 ingress
```

- a. Add drop rule.

```
# tc filter add block 22 protocol arp parent ffff: prio 3 \
    flower \
    dst_mac e4:11:22:11:4a:51 \
    action drop
```

- b. Add redirect rule from bond to representor.

```
# tc filter add block 22 protocol arp parent ffff: prio 3 \
    flower \
    dst_mac e4:11:22:11:4a:50 \
    action mirred egress redirect dev ens4f0_0
```

- c. Add redirect rule from representor to bond.

```
# tc filter add dev ens4f0_0 protocol arp parent ffff: prio 3 \
    flower \
    dst_mac ec:0d:9a:8a:28:42 \
    action mirred egress redirect dev bond0
```

2. Without shared block (supported from kernel 4.15 and below).

- a. Add redirect rule from bond to representor.

```
# tc filter add dev bond0 protocol arp parent ffff: prio 1 \
    flower \
    dst_mac e4:11:22:11:4a:50 \
    action mirred egress redirect dev ens4f0_0
```

- b. Add redirect rule from representor to bond.

```
# tc filter add dev ens4f0_0 protocol arp parent ffff: prio 3 \
    flower \
    dst_mac ec:0d:9a:8a:28:42 \
    action mirred egress redirect dev bond0
```

2.3.6.5.1.7 Classification Fields (Matches)

OVS-Kernel supports multiple classification fields which packets can fully or partially match.

Ethernet Layer 2

- Destination MAC

- Source MAC
- Ethertype

Supported on all kernels.

In OVS dump flows:

```
skb_priority(0/0),skb_mark(0/0),in_port(eth6),eth(src=00:02:10:40:10:0d
,dst=68:54:ed:00:af:de),eth_type(0x8100), packets:1981, bytes:206024, used:0.440s, dp:tc, actions:eth7
```

Using TC rules:

```
tc filter add dev $rep parent ffff: protocol arp pref 1 \
flower \
dst_mac e4:1d:2d:5d:25:35 \
src_mac e4:1d:2d:5d:25:34 \
action mirred egress redirect dev $NIC
```

IPv4/IPv6

- Source address
- Destination address
- Protocol
 - TCP/UDP/ICMP/ICMPv6
- TOS
- TTL (HLIMIT)

Supported on all kernels.

In OVS dump flows:

```
IPv4:
ipv4(src=0.0.0.0/0.0.0.0,dst=0.0.0.0/0.0.0.0,proto=17,tos=0/0,ttl=0/0,frag=no)
IPv6:
ipv6(src>::::,dst=1:1::3:1040:1008,label=0/0,proto=58,tclass=0/0x3,hlimit=64),
```

Using TC rules:

```
IPv4:
tc filter add dev $rep parent ffff: protocol ip pref 1 \
flower \
dst_ip 1.1.1.1 \
src_ip 1.1.1.2 \
ip_proto TCP \
ip_tos 0x3 \
ip_ttl 63 \
action mirred egress redirect dev $NIC

IPv6:
tc filter add dev $rep parent ffff: protocol ipv6 pref 1 \
flower \
dst_ip 1:1::3:1040:1009 \
src_ip 1:1::3:1040:1008 \
ip_proto TCP \
ip_tos 0x3 \
ip_ttl 63 \
action mirred egress redirect dev $NIC
```

TCP/UDP Source and Destination ports & TCP Flags

- TCP/UDP source and destinations ports
- TCP flags

Supported kernels are kernel > 4.13 and RHEL > 7.5

In OVS dump flows:

```
TCP: tcp(src=0/0,dst=32768/0x8000),
UDP: udp(src=0/0,dst=32768/0x8000),
TCP flags: tcp_flags(0/0)
```

Using TC rules:

```
tc filter add dev $rep parent ffff: protocol ip pref 1 \
flower \
ip_proto TCP \
dst_port 100 \
src_port 500 \
tcp_flags 0x4/0x7 \
action mirred egress redirect dev $NIC
```

VLAN

- ID
- Priority
- Inner vlan ID and Priority

Supported kernels: All (QinQ: kernel 4.19 and higher, and RHEL 7.7 and higher)

In OVS dump flows:

```
eth_type(0x8100),vlan(vid=2347,pcp=0),
```

Using TC rules:

```
tc filter add dev $rep parent ffff: protocol 802.1Q pref 1 \
      flower \
      vlan_ethertype 0x800 \
      vlan_id 100 \
      vlan_prio 0 \
      action mirred egress redirect dev $NIC
QinQ:
tc filter add dev $rep parent ffff: protocol 802.1Q pref 1 \
      flower \
      vlan_ethertype 0x8100 \
      vlan_id 100 \
      vlan_prio 0 \
      cvlan_id 20 \
      cvlan_prio 0 \
      cvlan_ethertype 0x800 \
      action mirred egress redirect dev $NIC
```

Tunnel

- ID (Key)
- Source IP address
- Destination IP address
- Destination port
- TOS (supported from kernel 4.19 and above & RHEL 7.7 and above)
- TTL (support from kernel 4.19 and above & RHEL 7.7 and above)
- Tunnel options (Geneve)

Supported kernels:

- VXLAN: All
- GRE: Kernel > 5.0, RHEL 7.7 and above
- Geneve: Kernel > 5.0, RHEL 7.7 and above

In OVS dump flows:

```
tunnel(tun_id=0x5,src=121.9.1.1,dst=131.10.1.1,ttl=0/0,tp_dst=4789,flags(+key))
```

Using TC rules:

```
# tc filter add dev $rep protocol 802.1Q parent ffff: pref 1
flower \
vlan_ethertype 0x800 \
vlan_id 100 \
vlan_prio 0 \
action mirred egress redirect dev $NIC
QinQ:
# tc filter add dev vxlan100 protocol ip parent ffff: \
flower \
    skip_sw \
    dst_mac e4:11:22:11:4a:51 \
    src_mac e4:11:22:11:4a:50 \
    enc_src_ip 20.1.11.1 \
    enc_dst_ip 20.1.12.1 \
    enc_key_id 100 \
    enc_dst_port 4789 \
    action tunnel_key unset \
    action mirred egress redirect dev ens4f0_0
```

2.3.6.5.1.8 Supported Actions

Forward

Forward action allows for packet redirection:

- From VF to wire
- Wire to VF
- VF to VF

Supported on all kernels.

In OVS dump flows:

```
skb_priority(0/0),skb_mark(0/0),in_port(eth6),eth(src=00:02:10:40:10:0d
,dst=68:54:ed:00:af:de),eth_type(0x8100), packets:1981, bytes:206024, used:0.440s, dp:tc, actions:eth7
```

Using TC rules:

```
tc filter add dev $rep parent ffff: protocol arp pref 1 \
flower \
    dst_mac e4:1d:2d:5d:25:35 \
    src_mac e4:1d:2d:5d:25:34 \
    action mirred egress redirect dev $NIC
```

Drop

Drop action allows to drop incoming packets.

Supported on all kernels.

In OVS dump flows:

```
skb_priority(0/0),skb_mark(0/0),in_port(eth6),eth(src=00:02:10:40:10:0d
,dst=68:54:ed:00:af:de),eth_type(0x8100), packets:1981, bytes:206024, used:0.440s, dp:tc, actions:drop
```

Using TC rules:

```
tc filter add dev $rep parent ffff: protocol arp pref 1 \
    flower \
        dst_mac e4:1d:2d:5d:25:35 \
        src_mac e4:1d:2d:5d:25:34 \
    action drop
```

Statistics

By default, each flow collects the following statistics:

- Packets - number of packets which hit the flow
- Bytes - total number of bytes which hit the flow
- Last used - the amount of time passed since last packet hit the flow

Supported on all kernels.

In OVS dump flows:

```
skb_priority(0/0),skb_mark(0/0),in_port(eth6),eth(src=00:02:10:40:10:0d
,dst=68:54:ed:00:af:de),eth_type(0x8100), packets:1981, bytes:206024, used:0.440s, dp:tc, actions:drop
```

Using TC rules:

```
#tc -s filter show dev $rep ingress

filter protocol ip pref 2 flower chain 0
filter protocol ip pref 2 flower chain 0 handle 0x2
eth_type ipv4
ip_proto tcp
src_ip 192.168.140.100
src_port 80
skip_sw
in_hw
  action order 1: mirrored (Egress Redirect to device p0v11_r) stolen
  index 34 ref 1 bind 1 installed 144 sec used 0 sec
Action statistics:
Sent 388344 bytes 2942 pkt (dropped 0, overlimits 0 requeues 0)
backlog 0b 0p requeues 0
```

Tunnels (Encapsulation/Decapsulation)

OVS-kernel supports offload of tunnels using encapsulation and decapsulation actions.

- Encapsulation - pushing of tunnel header is supported on Tx
- Decapsulation - popping of tunnel header is supported on Rx

Supported Tunnels:

- VXLAN (IPv4/IPv6) - supported on all Kernels
- GRE (IPv4/IPv6) - supported on kernel 5.0 and above & RHEL 7.6 and above
- Geneve (IPv4/IPv6) - supported on kernel 5.0 and above & RHEL 7.6 and above

OVS configuration:

In case of offloading tunnel, the PF/bond should not be added as a port in the OVS datapath. It should rather be assigned with the IP address to be used for encapsulation.

The example below shows two hosts (PFs) with IPs 1.1.1.177 and 1.1.1.75, where the PF device on both hosts is enp4s0f0, and the VXLAN tunnel is set with VNID 98:

- On the first host:

```
# ip addr add 1.1.1.177/24 dev enp4s0f1
# ovs-vsctl add-port ovs-sriov vxlan0 -- set interface vxlan0 type=vxlan
options:local_ip=1.1.1.177 options:remote_ip=1.1.1.75 options:key=98
```

- On the second host:

```
# ip addr add 1.1.1.75/24 dev enp4s0f1
# ovs-vsctl add-port ovs-sriov vxlan0 -- set interface vxlan0 type=vxlan
options:local_ip=1.1.1.75 options:remote_ip=1.1.1.177 options:key=98
• for GRE IPv4 tunnel need use type=gre
• for GRE IPv6 tunnel need use type=ip6gre
• for GENEVE tunnel need use type=geneve
```

When encapsulating guest traffic, the VF's device MTU must be reduced to allow the host/HW to add the encap headers without fragmenting the resulted packet. As such, the VF's MTU must be lowered by 50 bytes from the uplink MTU for IPv4 and 70 bytes for IPv6.

Tunnel offload using TC rules:

```
Encapsulation:
# tc filter add dev ens4f0_0 protocol 0x806 parent ffff: \
  flower \
    skip_sw \
    dst_mac e4:11:22:11:4a:51 \
    src_mac e4:11:22:11:4a:50 \
    action tunnel_key set \
    src_ip 20.1.12.1 \
    dst_ip 20.1.11.1 \
    id 100 \
    action mirred egress redirect dev vxlan100

Decapsulation:
# tc filter add dev vxlan100 protocol 0x806 parent ffff: \
  flower \
    skip_sw \
    dst_mac e4:11:22:11:4a:51 \
    src_mac e4:11:22:11:4a:50 \
    enc_src_ip 20.1.11.1 \
    enc_dst_ip 20.1.12.1 \
    enc_key_id 100 \
    enc_dst_port 4789 \
    action tunnel_key unset \
    action mirred egress redirect dev ens4f0_0
```

VLAN Push/Pop

OVS-kernel supports offload of vlan header push/pop actions.

- Push—pushing of VLAN header is supported on Tx
- Pop—popping of tunnel header is supported on Rx

Starting with ConnectX-6 Dx hardware models and above, pushing of VLAN header is also supported on Rx, and popping of VLAN header is also supported on Tx.

OVS Configuration

Add a tag=\$TAG section for the OVS command line that adds the representor ports. For example, VLAN ID 52 is being used here.

```
# ovs-vsctl add-port ovs-sriov enp4s0f0
# ovs-vsctl add-port ovs-sriov enp4s0f0_0 tag=52
# ovs-vsctl add-port ovs-sriov enp4s0f0_1 tag=52
```

The PF port should not have a VLAN attached. This will cause OVS to add VLAN push/pop actions when managing traffic for these VFs.

Dump Flow Example

```
recirc_id(0),in_port(3),eth(src=e4:11:22:33:44:50,dst=00:02:c9:e9:bb:b2),eth_type(0x0800),ipv4(frag=no), \
packets:0, bytes:0, used:never, actions:push_vlan(vid=52,pcp=0),2
recirc_id(0),in_port(2),eth(src=00:02:c9:e9:bb:b2,dst=e4:11:22:33:44:50),eth_type(0x8100), \
vlan(vid=52,pcp=0),encap(eth_type(0x0800),ipv4(frag=no)), packets:0, bytes:0, used:never, actions:pop_vlan,3
```

VLAN Offload using TC Rules Example

```
# tc filter add dev ens4f0_0 protocol ip parent ffff: \
flower \
    skip_sw \
    dst_mac e4:11:22:11:4a:51 \
    src_mac e4:11:22:11:4a:50 \
    action vlan push id 100 \
    action mirred egress redirect dev ens4f0
# tc filter add dev ens4f0 protocol 802.1Q parent ffff: \
flower \
    skip_sw \
    dst_mac e4:11:22:11:4a:51 \
    src_mac e4:11:22:11:4a:50 \
    vlan_ethertype 0x800 \
    vlan_id 100 \
    vlan_prio 0 \
    action vlan pop \
    action mirred egress redirect dev ens4f0_0
```

TC Configuration for ConnectX-6 Dx and Above

Example of VLAN Offloading with popping header on Tx and pushing on Rx using TC Rules:

```
# tc filter add dev ens4f0_0 ingress protocol 802.1Q parent ffff: \
flower \
    vlan_id 100 \
    action vlan pop \
    action tunnel_key set \
    src_ip 4.4.4.1 \
    dst_ip 4.4.4.2 \
    dst_port 4789 \
    id 42 \
    action mirred egress redirect dev vxlan0
# tc filter add dev vxlan0 ingress protocol all parent ffff: \
flower \
    enc_dst_ip 4.4.4.1 \
    enc_src_ip 4.4.4.2 \
    enc_dst_port 4789 \
    enc_key_id 42 \
    action tunnel_key unset \
    action vlan push id 100 \
    action mirred egress redirect dev ens4f0_0
```

Header Rewrite

This action allows for modifying packet fields.

Ethernet Layer 2

- Destination MAC
- Source MAC

Supported kernels: Kernel 4.14 and above & RHEL 7.5 and above

In OVS dump flows:

```
skb_priority(0/0),skb_mark(0/0),in_port(eth6),eth(src=00:02:10:40:10:0d
,dst=68:54:ed:00:af:de),eth_type(0x8100), packets:1981, bytes:206024, used:0.440s, dp:tc, actions: set(eth(src=68:5
4:ed:00:f4:ab,dst=fa:16:3e:dd:69:c4)),eth7
```

Using TC rules:

```
tc filter add dev $rep parent ffff: protocol arp pref 1 \
    flower \
        dst_mac e4:1d:2d:5d:25:35 \
        src_mac e4:1d:2d:5d:25:34 \
    action pedit ex \
        munge eth dst set 20:22:33:44:55:66 \
        munge eth src set aa:ba:cc:dd:ee:fe \
        action mirred egress redirect dev $NIC
```

IPv4/IPv6

- Source address
- Destination address
- Protocol
- TOS
- TTL (HLIMIT)

Supported kernels: Kernel 4.14 and above & RHEL 7.5 and above

In OVS dump flows:

```
IPv4:
  set (eth(src=de:e8:ef:27:5e:45,dst=00:00:01:01:01:01)),
  set (ipv4(src=10.10.0.111,dst=10.20.0.122,ttl=63))
IPv6:
  set (ipv6(dst=2001:1:6::92eb:fcbe:flc8,hlimit=63)),
```

Using TC rules:

```
IPv4:
tc filter add dev $rep parent ffff: protocol ip pref 1 \
    flower \
        dst_ip 1.1.1.1 \
        src_ip 1.1.1.2 \
        ip_proto TCP \
        ip_tos 0x3 \
        ip_ttl 63 \
    pedit ex \
        munge ip src set 2.2.2.1 \
        munge ip dst set 2.2.2.2 \
        munge ip tos set 0 \
        munge ip ttl dec \
        action mirred egress redirect dev $NIC

IPv6:
tc filter add dev $rep parent ffff: protocol ipv6 pref 1 \
    flower \
        dst_ip 1:1:1::3:1040:1009 \
        src_ip 1:1:1::3:1040:1008 \
        ip_proto tcp \
        ip_tos 0x3 \
        ip_ttl 63 \
    pedit ex \
        munge ipv6 src set 2:2:2::3:1040:1009 \
        munge ipv6 dst set 2:2:2::3:1040:1008 \
        munge ipv6 hlimit dec \
        action mirred egress redirect dev $NIC
```

IPv4 and IPv6 header rewrite is only supported with match on UDP/TCP/ICMP protocols.

TCP/UDP Source and Destination Ports

- TCP/UDP source and destinations ports

Supported kernels: kernel > 4.16 & RHEL > 7.6

In OVS dump flows:

```
TCP:
    set (tcp(src= 32768/0xffff,dst=32768/0xffff)),
UDP:
    set (udp(src= 32768/0xffff,dst=32768/0xffff)),
```

Using TC rules:

```
TCP:
    tc filter add dev $rep parent ffff: protocol ip pref 1 \
        flower \
        dst_ip 1.1.1.1 \
        src_ip 1.1.1.2 \
        ip_proto tcp \
        ip_tos 0x3 \
        ip_ttl 63 \
        pedit ex \
        pedit ex munge ip tcp sport set 200
        pedit ex munge ip tcp dport set 200
        action mirred egress redirect dev $NIC
UDP:
    tc filter add dev $rep parent ffff: protocol ip pref 1 \
        flower \
        dst_ip 1.1.1.1 \
        src_ip 1.1.1.2 \
        ip_proto udp \
        ip_tos 0x3 \
        ip_ttl 63 \
        pedit ex \
        pedit ex munge ip udp sport set 200
        pedit ex munge ip udp dport set 200
        action mirred egress redirect dev $NIC
```

VLAN

- ID

Supported on all kernels.

In OVS dump flows:

```
Set (vlan(vid=2347,pcp=0/0)),
```

Using TC rules:

```
tc filter add dev $rep parent ffff: protocol 802.1Q pref 1 \
    flower \
    vlan_ethtype 0x800 \
    vlan_id 100 \
    vlan_prio 0 \
    action vlan modify id 11 pipe
    action mirred egress redirect dev $NIC
```

Connection Tracking

The TC connection tracking action performs connection tracking lookup by sending the packet to netfilter conntrack module. Newly added connections may be associated, via the ct commit action, with a 32 bit mark, 128 bit label and source/destination NAT values.

The following example allows ingress tcp traffic from the uplink representor to vf1_rep, while assuring that egress traffic from vf1_rep is only allowed on established connections. In addition, mark and source IP NAT is applied.

In OVS dump flows:

```
ct (zone=2,nat)
```



```
ct_state(+est+trk)
actions:ct (commit,zone=2,mark=0x4/0xffffffff,nat(src=5.5.5.5))
```

Using TC rules:

```
# tc filter add dev $uplink_rep ingress chain 0 prio 1 proto ip \
    flower \
    ip_proto tcp \
    ct_state -trk \
    action ct zone 2 nat pipe \
    action goto chain 2
# tc filter add dev $uplink_rep ingress chain 2 prio 1 proto ip \
    flower \
    ct_state +trk+new \
    action ct zone 2 commit mark 0xbb nat src addr 5.5.5.7 pipe \
    action mirrored egress redirect dev $vfl_rep
# tc filter add dev $uplink_rep ingress chain 2 prio 1 proto ip \
    flower \
    ct_zone 2 \
    ct_mark 0xbb \
    ct_state +trk+est \
    action mirrored egress redirect dev $vfl_rep

#Setup filters on $vfl_rep, allowing only established connections of zone 2 through, and reverse nat (dst nat
in this case)
# tc filter add dev $vfl_rep ingress chain 0 prio 1 proto ip \
    flower \
    ip_proto tcp \
    ct_state -trk \
    action ct zone 2 nat pipe \
    action goto chain 1
# tc filter add dev $vfl_rep ingress chain 1 prio 1 proto ip \
    flower \
    ct_zone 2 \
    ct_mark 0xbb \
    ct_state +trk+est \
    action mirrored egress redirect dev eth0
```

Connection Tracking Performance Tuning

- Max offloaded connections—specifies the limit on the number of offloaded connections.

Example:

```
# devlink dev param set pci/${pci_dev} name ct_max_offloaded_conns value $max cmode runtime
```

- Allow mixed NAT/non-NAT CT—allows offloading of the following scenario:

```
• cookie=0x0, duration=21.843s, table=0, n_packets=4838718, n_bytes=241958846, ct_state=-
trk,ip,in_port=enp8s0f0 actions=ct (table=1,zone=2)
• cookie=0x0, duration=21.823s, table=1, n_packets=15363, n_bytes=773526, ct_state=new+trk,ip,in_port=en
p8s0f0 actions=ct (commit,zone=2,nat(dst=11.11.11.11)),output:"enp8s0f0_1"
• cookie=0x0, duration=21.806s, table=1, n_packets=4767594, n_bytes=238401190,
ct_state=est+trk,ip,in_port=enp8s0f0 actions=ct (zone=2,nat),output:"enp8s0f0_1"
```

Example:

```
# echo enable > /sys/class/net/<device>/compat/devlink/ct_action_on_nat_conns
```

Forward to Chain (TC Only)

TC interface supports adding flows on different chains. Only chain 0 is accessed by default. Access to the other chains requires use of the goto action.

In this example, a flow is created on chain 1 without any match and redirect to wire.

The second flow is created on chain 0 and match on source MAC and action goto chain 1.

This example simulates simple MAC spoofing.

```
#tc filter add dev $rep parent ffff: protocol all chain 1 pref 1 \
    flower \
    action mirrored egress redirect dev $NIC
#tc filter add dev $rep parent ffff: protocol all chain 1 pref 1 \
```

```

flower \
src_mac aa:bb:cc:aa:bb:cc
action goto chain 1

```

2.3.6.5.1.9 Port Mirroring (Flow Based VF Traffic Mirroring for ASAP²)

Unlike para-virtual configurations, when the VM traffic is offloaded to the hardware via SR-IOV VF, the host side Admin cannot snoop the traffic (e.g. for monitoring).

ASAP² uses the existing mirroring support in OVS and TC along with the enhancement to the offloading logic in the driver to allow mirroring the VF traffic to another VF.

The mirrored VF can be used to run traffic analyzer (tcpdump, wireshark, etc) and observe the traffic of the VF being mirrored.

The example below shows the creation of port mirror on the following configuration:

```

# ovs-vsctl show
09d8a574-9c39-465c-9f16-47d81c12f88a
Bridge br-vxlan
  Port "enp4s0f0_1"
    Interface "enp4s0f0_1"
  Port "vxlan0"
    Interface "vxlan0"
      type: vxlan
      options: {key="100", remote_ip="192.168.1.14"}
  Port "enp4s0f0_0"
    Interface "enp4s0f0_0"
  Port "enp4s0f0_2"
    Interface "enp4s0f0_2"
  Port br-vxlan
    Interface br-vxlan
      type: internal
ovs_version: "2.14.1"

```

- To set enp4s0f0_0 as the mirror port, and mirror all of the traffic:

```

# ovs-vsctl -- --id=@p get port enp4s0f0_0 \
-- --id=@m create mirror name=m0 select-all=true output-port=@p \
-- set bridge br-vxlan mirrors=@m

```

- To set enp4s0f0_0 as the mirror port, and only mirror the traffic, the destination is enp4s0f0_1:

```

# ovs-vsctl -- --id=@p1 get port enp4s0f0_0 \
-- --id=@p2 get port enp4s0f0_1 \
-- --id=@m create mirror name=m0 select-dst-port=@p2 output-port=@p1 \
-- set bridge br-vxlan mirrors=@m

```

- To set enp4s0f0_0 as the mirror port, and only mirror the traffic the source is enp4s0f0_1:

```

# ovs-vsctl -- --id=@p1 get port enp4s0f0_0 \
-- --id=@p2 get port enp4s0f0_1 \
-- --id=@m create mirror name=m0 select-src-port=@p2 output-port=@p1 \
-- set bridge br-vxlan mirrors=@m


```

- To set enp4s0f0_0 as the mirror port and mirror, all the traffic on enp4s0f0_1:

```

# ovs-vsctl -- --id=@p1 get port enp4s0f0_0 \
-- --id=@p2 get port enp4s0f0_1 \
-- --id=@m create mirror name=m0 select-dst-port=@p2 select-src-port=@p2 output-port=@p1 \
-- set bridge br-vxlan mirrors=@m

```

 To clear the mirror port:

```
# ovs-vsctl clear bridge br-vxlan mirrors
```

Mirroring using TC:

```
Mirror to VF
tc filter add dev $rep parent ffff: protocol arp pref 1 \
    flower \
    dst_mac e4:1d:2d:5d:25:35 \
    src_mac e4:1d:2d:5d:25:34 \
    action mirred egress mirror dev $mirror_rep pipe \
    action mirred egress redirect dev $NIC

Mirror to tunnel:
tc filter add dev $rep parent ffff: protocol arp pref 1 \
    flower \
    dst_mac e4:1d:2d:5d:25:35 \
    src_mac e4:1d:2d:5d:25:34 \
    action tunnel_key set \
    src_ip 1.1.1.1 \
    dst_ip 1.1.1.2 \
    dst_port 4789 \
    id 768 \
    pipe \
    action mirred egress mirror dev vxlan100 pipe \
    action mirred egress redirect dev $NIC
```

2.3.6.5.1.10 Forward to Multiple Destinations

Forward to up to 32 destinations (representors and tunnels) is supported using TC.

Example 1: forward to 32 VFs.

```
tc filter add dev $NIC parent ffff: protocol arp pref 1 \
    flower \
    dst_mac e4:1d:2d:5d:25:35 \
    src_mac e4:1d:2d:5d:25:34 \
    action mirred egress mirror dev $rep0 pipe \
    action mirred egress mirror dev $rep1 pipe \
    ...
    action mirred egress mirror dev $rep30 pipe \
    action mirred egress redirect dev $rep31
```

Example 2: forward to 16 tunnels.

```
tc filter add dev $rep parent ffff: protocol arp pref 1 \
    flower \
    dst_mac e4:1d:2d:5d:25:35 \
    src_mac e4:1d:2d:5d:25:34 \
    action tunnel_key set src_ip $ip_src dst_ip $ip_dst \
dst_port 4789 id 0 nocsum \
pipe action mirred egress mirror dev vxlan0 pipe \
    action tunnel_key set src_ip $ip_src dst_ip $ip_dst \
dst_port 4789 id 1 nocsum \
pipe action mirred egress mirror dev vxlan0 pipe \
...
    action tunnel_key set src_ip $ip_src dst_ip $ip_dst \
dst_port 4789 id 15 nocsum \
pipe action mirred egress redirect dev vxlan0
```

- TC supports up to 32 actions
- If header rewrite is used, then all destinations should have the same header rewrite
- If VLAN push/pop is used, then all destinations should have the same VLAN ID and actions

2.3.6.5.1.11 sFLOW

This feature allows for monitoring traffic sent between two VMs on the same host using an sFlow collector.

The example below assumes the environment is configured as described below.

```
# ovs-vsctl show
09d8a574-9c39-465c-9f16-47d81c12f88a
Bridge br-vxlan
  Port "enp4s0f0_1"
    Interface "enp4s0f0_1"
  Port "vxlan0"
    Interface "vxlan0"
      type: vxlan
      options: {key="100", remote_ip="192.168.1.14"}
  Port "enp4s0f0_0"
    Interface "enp4s0f0_0"
  Port "enp4s0f0_2"
    Interface "enp4s0f0_2"
  Port br-vxlan
    Interface br-vxlan
      type: internal
ovs_version: "2.14.1"
```

To sample all traffic over the OVS bridge:

```
# ovs-vsctl -- --id=@sflow create sflow agent="\$SFLOW_AGENT" \
target="\$SFLOW_TARGET:\$SFLOW_PORT" header=\$SFLOW_HEADER \
sampling=\$SFLOW_SAMPLING polling=10 \
-- set bridge br-vxlan sflow=@sflow
```

Parameter	Description
SFLOW_AGENT	Indicates that the sFlow agent should send traffic from SFLOW_AGENT's IP address
SFLOW_TARGET	Remote IP address of the sFLOW collector
SFLOW_HEADER	Size of packet header to sample (in bytes)
SFLOW_SAMPLING	Sample rate

To clear the sFLOW configuration:

```
# ovs-vsctl clear bridge br-vxlan sflow
```

To list the sFLOW configuration:

```
# ovs-vsctl list sflow
```

sFLOW using TC:

```
Sample to VF
tc filter add dev $rep parent ffff: protocol arp pref 1 \
  flower \
  dst_mac e4:1d:2d:5d:25:35 \
  src_mac e4:1d:2d:5d:25:34 \
  action sample rate 10 group 5 trunc 96 \
  action mirred egress redirect dev $NIC
```

Userspace application is needed in order to process to sampled packet from the kernel.
 Example: <https://github.com/Mellanox/libpsample>

2.3.6.5.1.12 Rate Limit

OVS-kernel supports offload of VF rate limit using OVS configuration and TC.

The example below sets a rate limit to the VF related to representor eth0 to 10Mbps.

```
OVS:
# ovs-vsctl set interface eth0 ingress_policing_rate=10000

tc:
# tc_filter add dev eth0 root prio 1 protocol ip matchall skip_sw action police rate 10mbit burst 20k
```

2.3.6.5.1.13 Kernel Requirements

This kernel config should be enabled in order to support switchdev offload.

- CONFIG_NET_ACT_CSUM - needed for action csum
- CONFIG_NET_ACT_PEDIT - needed for header rewrite
- CONFIG_NET_ACT_MIRRED - needed for basic forward
- CONFIG_NET_ACT_CT - needed for connection tracking (supported from kernel 5.6)
- CONFIG_NET_ACT_VLAN - needed for action vlan push/pop
- CONFIG_NET_ACT_GACT
- CONFIG_NET_CLS_FLOWER
- CONFIG_NET_CLS_ACT
- CONFIG_NET_SWITCHDEV
- CONFIG_NET_TC_SKB_EXT - needed for connection tracking (supported from kernel 5.6)
- CONFIG_NET_ACT_CT - needed for connection tracking (supported from kernel 5.6)
- CONFIG_NFT_FLOW_OFFLOAD
- CONFIG_NET_ACT_TUNNEL_KEY
- CONFIG_NF_FLOW_TABLE - needed for connection tracking (supported from kernel 5.6)
- CONFIG_SKB_EXTENSIONS - needed for connection tracking (supported from kernel 5.6)
- CONFIG_NET_CLS_MATCHALL
- CONFIG_NET_ACT_POLICE
- CONFIG_MLX5_ESWITCH

2.3.6.5.1.14 VF Metering

OVS-kernel supports offloading of VF metering (TX and RX) using sysfs. Metering of number of packets per second (PPS) and bytes per second (BPS) is supported.

The example bellow sets Rx meter on VF 0 with value 10Mbps BPS.

```
echo 10000000 > /sys/class/net/enp4s0f0/device/sriov/0/meters/rx/bps/rate
echo 65536 > /sys/class/net/enp4s0f0/device/sriov/0/meters/rx/bps/burst
```

The example bellow sets Tx meter on VF 0 with value 1000 PPS.

```
echo 1000 > /sys/class/net/enp4s0f0/device/sriov/0/meters/tx/pps/rate
echo 100 > /sys/class/net/enp4s0f0/device/sriov/0/meters/tx/pps/burst
```

Both rate and burst must not be zero and burst may need to be adjusted according to the requirements.

The following counters can be used to query the number dropped packet/bytes:

```
#cat /sys/class/net/enp8s0f0/device/sriov/0/meters/rx/pps/packets_dropped
#cat /sys/class/net/enp8s0f0/device/sriov/0/meters/rx/pps/bytes_dropped
#cat /sys/class/net/enp8s0f0/device/sriov/0/meters/rx/bps/packets_dropped
#cat /sys/class/net/enp8s0f0/device/sriov/0/meters/rx/bps/bytes_dropped
#cat /sys/class/net/enp8s0f0/device/sriov/0/meters/tx/pps/packets_dropped
#cat /sys/class/net/enp8s0f0/device/sriov/0/meters/tx/pps/bytes_dropped
#cat /sys/class/net/enp8s0f0/device/sriov/0/meters/tx/bps/packets_dropped
#cat /sys/class/net/enp8s0f0/device/sriov/0/meters/tx/bps/bytes_dropped
```

2.3.6.5.1.15 Representor Metering

Metering for uplink and VF representors traffic support has been added.

Traffic going to a representor device can be a result of a miss in the embedded switch (eSwitch) FDB tables. This means that a packet which arrived from that representor into the eSwitch was not matched against the existing rules in the hardware FDB tables and needs to be forwarded to software to be handled there and is, therefore, forwarded to the originating representor device driver.

The meter allows to configure the max rate [packets/sec] and max burst [packets] for traffic going to the representor driver. Any traffic exceeding values provided by the user will be dropped in hardware. There are statistics that show number of dropped packets.

The configuration of a representors metering is done via a new sysfs called `miss_rl_cfg`.

- Full path of the `miss_rl_cfg` parameter: `/sys/class/net//rep_config/miss_rl_cfg`
- Usage: `echo "<rate> <burst>" > /sys/class/net//rep_config/miss_rl_cfg`. Rate is the max rate of packets allowed for this representor (in packets/sec units) and burst is the max burst size allowed for this representor (in packets units). Both values must be specified. The default is 0 for both, meaning unlimited rate and burst.

To view the amount of packets and bytes that were dropped due to traffic exceeding the user-provided rate and burst, two read-only sysfs for statistics are exposed.

- `/sys/class/net//rep_config/miss_rl_dropped_bytes` counts how many FDB-miss bytes were dropped due to reaching the miss limits
- `/sys/class/net//rep_config/miss_rl_dropped_packets` counts how many FDB-miss packets were dropped due to reaching the miss limits

2.3.6.5.1.16 Open vSwitch Metering

There are two types of meters, kpps (kilobits per second) and pktpps (packets per second), which are described in Meter Syntax of OpenFlow 1.3+ Switch Meter Table Commands. OVS-Kernel supports offloading them both.

The example below is to offload a kpps meter. Please follow the steps after doing basic configurations as described in section 5.1.3.

1. Create OVS meter with a target rate.

```
ovs-ofctl -O OpenFlow13 add-meter ovs-sriov meter=1,kbps,band=type=drop,rate=204800
```

2. Delete the default rule.

```
ovs-ofctl del-flows ovs-sriov
```

3. Configure OpenFlow rules. Here VF bandwidth on the receiving side will be limited by the rate configured in step 1.

```
ovs-ofctl -O OpenFlow13 add-flow ovs-sriov 'ip,d1_dst=e4:11:22:33:44:50,actions=
meter:1,output:enp4s0f0_0'
ovs-ofctl -O OpenFlow13 add-flow ovs-sriov 'ip,d1_src=e4:11:22:33:44:50,actions= output:enp4s0f0'
ovs-ofctl -O OpenFlow13 add-flow ovs-sriov 'arp,actions=normal'
```

4. Run iperf server and be ready to receive UDP traffic. On the outer node, run iperf client to send UDP traffic to this VF. After traffic starts, check the offloaded meter rule.

```
ovs-appctl dpctl/dump-flows --names type=offloaded
recirc_id(0),in_port(enp4s0f0),eth(dst=e4:11:22:33:44:50),eth_type(0x0800),ipv4(frag=no), packets:11626587,
bytes:17625889188, used:0.470s, actions:meter(0),enp4s0f0_0
```

In order to verify metering, iperf client should set the target bandwidth with a number which is larger than the meter rate configured. Then it will be visible that packets are received with the limited rate on the server side and the extra packets are dropped by hardware.

2.3.6.5.1.17 Multiport eSwitch Mode

The multiport eSwitch mode allows to add rules on a VF representor with an action forwarding the packet to the physical port of the physical function. This can be used to implement failover or forward packets based on external information such the cost of the route.

1. To configure this more, the nvconig parameter LAG_RESOURCE_ALLOCATION must be set.
2. After the driver loads, configure multiport eSwitch for each PF where enp8s0f0 and enp8s0f1 represent the netdevices for the PFs.

```
echo multiport_esw > /sys/class/net/enp8s0f0/compat/devlink/lag_port_select_mode
echo multiport_esw > /sys/class/net/enp8s0f1/compat/devlink/lag_port_select_mode
```

The mode becomes operational after entering switchdev mode on both PFs.

Rule example:

```
tc filter add dev enp8s0f0_0 prot ip root flower dst_ip 7.7.7.7 action mirred egress redirect dev enp8s0f1
```

2.3.6.5.2 OVS-DPDK Hardware Offloads

2.3.6.5.2.1 OVS-DPDK Hardware Offloads Configuration

 To configure OVS-DPDK HW offloads:

1. Unbind the VFs.

```
echo 0000:04:00.2 > /sys/bus/pci/drivers/mlx5_core/unbind  
echo 0000:04:00.3 > /sys/bus/pci/drivers/mlx5_core/unbind
```

Note: VMs with attached VFs must be powered off to be able to unbind the VFs.

2. Change the e-switch mode from Legacy to SwitchDev on the PF device (make sure all VFs are unbound). This will also create the VF representor netdevices in the host OS.

```
echo switchdev > /sys/class/net/enp4s0f0/compat/devlink/mode
```

To revert to SR-IOV Legacy mode:

```
echo legacy > /sys/class/net/enp4s0f0/compat/devlink/mode
```

Note that running this command will also result in the removal of the VF representor netdevices.

3. Bind the VFs.

```
echo 0000:04:00.2 > /sys/bus/pci/drivers/mlx5_core/bind  
echo 0000:04:00.3 > /sys/bus/pci/drivers/mlx5_core/bind
```

4. Run the Open vSwitch service.

```
systemctl start openvswitch
```

5. Enable hardware offload (disabled by default).

```
ovs-vsctl --no-wait set Open_vSwitch . other_config:dpdk-init=true  
ovs-vsctl set Open_vSwitch . other_config:hw-offload=true
```

6. Configure the DPDK white list.

```
ovs-vsctl --no-wait set Open_vSwitch . other_config:dpdk-extra="-a  
0000:01:00.0,representor=[0],dv_flow_en=1,dv_esw_en=1,dv_xmeta_en=1"
```

Representor=[0-N]

7. Restart the Open vSwitch service. This step is required for HW offload changes to take effect.

```
systemctl restart openvswitch
```

8. Create OVS-DPDK bridge.

```
ovs-vsctl --no-wait add-br br0-ovs -- set bridge br0-ovs datapath_type=netdev
```

9. Add PF to OVS.

```
ovs-vsctl add-port br0-ovs pf -- set Interface pf type=dpdk options:dpdk-devargs=0000:88:00.0
```

10. Add representor to OVS.


```
ovs-vsctl add-port br0-ovs representor -- set Interface representor type=dpdk options:dpdk-devargs=0000:88:00.0,representor=[0]
```

Representor=[0-N]

2.3.6.5.2.2 Offloading VXLAN Encapsulation/Decapsulation Actions

vSwitch in userspace rather than kernel-based Open vSwitch requires an additional bridge. The purpose of this bridge is to allow use of the kernel network stack for routing and ARP resolution.

The datapath needs to look-up the routing table and ARP table to prepare the tunnel header and transmit data to the output port.

Configuring VXLAN Encap/Decap Offloads

The configuration is done with:

- PF on 0000:03:00.0 PCI and MAC 98:03:9b:cc:21:e8
- Local IP 56.56.67.1 - br-phy interface will be configured to this IP
- Remote IP 56.56.68.1

➤ *To configure OVS-DPDK VXLAN:*

1. Create a br-phy bridge.

```
ovs-vsctl add-br br-phy -- set Bridge br-phy datapath_type=netdev -- br-set-external-id br-phy bridge-id br-phy -- set bridge br-phy fail-mode=standalone other_config:hwaddr=98:03:9b:cc:21:e8
```

2. Attach PF interface to br-phy bridge.

```
ovs-vsctl add-port br-phy p0 -- set Interface p0 type=dpdk options:dpdk-devargs=0000:03:00.0
```

3. Configure IP to the bridge.

```
ip addr add 56.56.67.1/24 dev br-phy
```

4. Create a br-ovs bridge.

```
ovs-vsctl add-br br-ovs -- set Bridge br-ovs datapath_type=netdev -- br-set-external-id br-ovs bridge-id br-ovs -- set bridge br-ovs fail-mode=standalone
```

5. Attach representor to br-ovs.

```
ovs-vsctl add-port br-ovs pf0vf0 -- set Interface pf0vf0 type=dpdk options:dpdk-devargs=0000:03:00.0,representor=[0]
```

6. Add a port for the VXLAN tunnel.

```
ovs-vsctl add-port ovs-sriov vxlan0 -- set interface vxlan0 type=vxlan options:local_ip=56.56.67.1 options:remote_ip=56.56.68.1 options:key=45 options:dst_port=4789
```

2.3.6.5.2.3 Connection Tracking Offload

Connection tracking enables stateful packet processing by keeping a record of currently open connections.

OVS flows using connection tracking can be accelerated using advanced Network Interface Cards (NICs) by offloading established connections.

To view offloaded connections, run:

```
ovs-appctl dpctl/offload-stats-show
```

2.3.6.5.2.4 SR-IOV VF LAG

 *To configure OVS-DPDK SR-IOV VF LAG:*

1. Enable SR-IOV on the NICs.

```
mlxconfig -d <PCI> set SRIOV_EN=1
```

2. Allocate the desired number of VFs per port.

```
echo $n > /sys/class/net/<net name>/device/sriov_numvfs
```

3. Unbind all VFs.

```
echo <VF PCI> >/sys/bus/pci/drivers/mlx5_core/unbind
```

4. Change both NICs' mode to SwitchDev.

```
devlink dev eswitch set pci/<PCI> mode switchdev
```

5. Create Linux bonding using kernel modules.

```
modprobe bonding mode=<desired mode>
```

Note: Other bonding parameters can be added here. The supported Bond modes are: Active-Backup, XOR and LACP.

6. Bring all PFs and VFs down.

```
ip link set <PF/VF> down
```

7. Attach both PFs to the bond.

```
ip link set <PF> master bond0
```

8. To work with VF-LAG with OVS-DPDK, add the bond master (PF) to the bridge.

```
ovs-vsctl add-port br-phy p0 -- set Interface p0 type=dpdk options:dpdk-devargs=0000:03:00.0 options:dpdk-lsc-interrupt=true
```

9. Add representor \$N of PF0 or PF1 to a bridge.

```

ovs-vsctl add-port br-phy rep$N -- set Interface rep$N type=dpmk options:dpmk-devargs=<PF0
PCI>,representor=pf0vf$N
OR
ovs-vsctl add-port br-phy rep$N -- set Interface rep$N type=dpmk options:dpmk-devargs=<PF0
PCI>,representor=pf1vf$N

```

2.3.6.5.2.5 VirtIO Acceleration through VF Relay (Software & Hardware vDPA)

Hardware vDPA is supported on ConnectX-6 Dx, ConnectX-6 Lx & BlueField-2 cards and above only.

Hardware vDPA is enabled by default. In case your hardware does not support vDPA, the driver will fall back to Software vDPA.

To check which vDPA mode is activated on your driver, run: `ovs-ofctl -O OpenFlow14 dump-ports br0-ovs` and look for `hw-mode` flag.

This feature has not been accepted to the OVS-DPDK Upstream yet, making its API subject to change.

In user space, there are two main approaches for communicating with a guest (VM), either through SR-IOV, or through virtIO.

Phy ports (SR-IOV) allow working with port representor, which is attached to the OVS and a matching VF is given with pass-through to the guest. HW rules can process packets from up-link and direct them to the VF without going through SW (OVS). Therefore, using SR-IOV achieves the best performance.

However, SR-IOV architecture requires the guest to use a driver specific to the underlying HW. Specific HW driver has two main drawbacks:

1. Breaks virtualization in some sense (guest is aware of the HW). It can also limit the type of images supported.
2. Gives less natural support for live migration.

Using virtIO port solves both problems. However, it reduces performance and causes loss of some functionalities, such as, for some HW offloads, working directly with virtIO. To solve this conflict, a new netdev type- `dpmkvdpa` has been created. The new netdev is similar to the regular DPDK netdev, yet introduces several additional functionalities.

`dpmkvdpa` translates between phy port to virtIO port. It takes packets from the Rx queue and sends them to the suitable Tx queue, and allows transfer of packets from virtIO guest (VM) to a VF, and vice-versa, benefitting from both SR-IOV and virtIO.

 To add vDPA port:

```

ovs-vsctl add-port br0 vdpao -- set Interface vdpao type=dpmkvdpa \
options:vdpao-socket-path=<sock path> \
options:vdpao-accelerator-devargs=<vf pci id> \
options:dpmk-devargs=<pf pci id>,representor=[id] \
options: vdpao-max-queues =<num queues> \
options: vdpao-sw=<true/false>

```

Note: vdp-max-queues is an optional field. When the user wants to configure 32 vDPA ports, the maximum queues number is limited to 8.

vDPA Configuration in OVS-DPDK Mode

Prior to configuring vDPA in OVS-DPDK mode, follow the steps below.

1. Generate the VF.

```
echo 0 > /sys/class/net/enp175s0f0/device/sriov_numvfs
echo 4 > /sys/class/net/enp175s0f0/device/sriov_numvfs
```

2. Unbind each VF.

```
echo <pci> > /sys/bus/pci/drivers/mlx5_core/unbind
```

3. Switch to SwitchDev mode.

```
echo switchdev >> /sys/class/net/enp175s0f0/compat/devlink/mode
```

4. Bind each VF.

```
echo <pci> > /sys/bus/pci/drivers/mlx5_core/bind
```

5. Initialize OVS with:

```
ovs-vsctl --no-wait set Open_vSwitch . other_config:dpdk-init=true
ovs-vsctl --no-wait set Open_vSwitch . other_config:hw-offload=true
```

➤ *To configure vDPA in OVS-DPDK mode on ConnectX-5 cards and above:*

1. Open vSwitch configuration.

```
ovs-vsctl --no-wait set Open_vSwitch . other_config:dpdk-extra="-a
0000:01:00.0,representor=[0],dv_flow_en=1,dv_esw_en=1,dv_xmeta_en=1"
/usr/share/openvswitch/scripts/ovs-ctl restart
```

2. Create OVS-DPDK bridge.

```
ovs-vsctl add-br br0-ovs -- set bridge br0-ovs datapath_type=netdev
ovs-vsctl add-port br0-ovs pf -- set Interface pf type=dpdk options:dpdk-devargs=0000:01:00.0
```

3. Create vDPA port as part of the OVS-DPDK bridge.

```
ovs-vsctl add-port br0-ovs vdp0 -- set Interface vdp0 type=dpdkvdp options:vdp-socket-path=/var/run/
virtio-forwarder/sock0 options:vdp-accelerator-devargs=0000:01:00.2 options:dpdk-devargs=0000:01:00.0,repr
esentor=[0] options: vdp-max-queues=8
```

➤ *To configure vDPA in OVS-DPDK mode on BlueField cards:*

Set the bridge with the software or hardware vDPA port:

- On the ARM side:
Create the OVS-DPDK bridge.

```
ovs-vsctl add-br br0-ovs -- set bridge br0-ovs datapath_type=netdev
ovs-vsctl add-port br0-ovs pf -- set Interface pf type=dpdk options:dpdk-devargs=0000:af:00.0
```

```
ovs-vsctl add-port br0-ovs rep -- set Interface rep type=dppk options:dppk-devargs=0000:af:00.0,representor=[0]
```

- On the host side:
Create the OVS-DPDK bridge.

```
ovs-vsctl add-br br1-ovs -- set bridge br1-ovs datapath_type=netdev protocols=OpenFlow14  
ovs-vsctl add-port br0-ovs vdp0 -- set Interface vdp0 type=dppkvdpa options:vdp0-socket-path=/var/run/  
virtio-forwarder/sock0 options:vdp0-accelerator-devargs=0000:af:00.2
```

Note: To configure SW vDPA, add "options:vdp0-sw=true" to the end of the command.

Software vDPA Configuration in OVS-Kernel Mode

SW vDPA can also be used in configurations where the HW offload is done through TC and not DPDK.

1. Open vSwitch configuration.

```
ovs-vsctl set Open_vSwitch . other_config:dppk-extra="-a  
0000:01:00.0,representor=[0],dv_flow_en=1,dv_esw_en=0,idv_xmeta_en=0,isolated_mode=1"  
/usr/share/openvswitch/scripts/ovs-ctl restart
```

2. Create OVS-DPDK bridge.

```
ovs-vsctl add-br br0-ovs -- set bridge br0-ovs datapath_type=netdev
```

3. Create vDPA port as part of the OVS-DPDK bridge.

```
ovs-vsctl add-port br0-ovs vdp0 -- set Interface vdp0 type=dppkvdpa options:vdp0-socket-path=/var/run/  
virtio-forwarder/sock0 options:vdp0-accelerator-devargs=0000:01:00.2 options:dppk-devargs=0000:01:00.0,repr  
esentor=[0] options: vdp0-max-queues=8
```

4. Create Kernel bridge.

```
ovs-vsctl add-br br-kernel
```

5. Add representors to Kernel bridge.

```
ovs-vsctl add-port br-kernel enpls0f0_0  
ovs-vsctl add-port br-kernel enpls0f0
```

2.3.6.5.2.6 Large MTU/Jumbo Frame Configuration

 To configure MTU/jumbo frames:

1. Verify that the Kernel version on the VM is 4.14 or above.

```
cat /etc/redhat-release
```

2. Set the MTU on both physical interfaces in the host.

```
ifconfig ens4f0 mtu 9216
```

3. Send a large size packet and verify that it is sent and received correctly.

```
tcpdump -i ens4f0 -nev icmp &  
ping 11.100.126.1 -s 9188 -M do -c 1
```

4. Enable `host_mtu` in xml, and add the following values to xml.

```
host_mtu=9216,csum=on,guest_csum=on,host_tso4=on,host_tso6=on
```

Example:

```
<qemu:commandline>
<qemu:arg value='-chardev' />
<qemu:arg value='socket,id=charnet1,path=/tmp/sock0,server' />
<qemu:arg value='-netdev' />
<qemu:arg value='vhost-user,chardev=charnet1,queues=16,id=hostnet1' />
<qemu:arg value='-device' />
<qemu:arg value='virtio-net-
pci,mq=on,vectors=34,netdev=hostnet1,id=net1,mac=00:21:21:24:02:01,bus=pci.0,addr=0xC,page-per-
vq=on,rx_queue_size=1024,tx_queue_size=1024,host_mtu=9216,csum=on,guest_csum=on,host_tso4=on,host_tso6=on' /
>
</qemu:commandline>
```

5. Add `mtu_request=9216` option to the OVS ports inside the container and restart the OVS:

```
ovs-vsctl add-port br0-ovs pf -- set Interface pf type=dpdk options:dpdk-devargs=0000:c4:00.0 mtu_request=9216
```

OR:

```
ovs-vsctl add-port br0-ovs vdpao -- set Interface vdpao type=dpdkvdpao options:vdpao-socket-path=/tmp/sock0
options:vdpao-accelerator-devargs=0000:c4:00.2 options:dpdk-devargs=0000:c4:00.0,representor=[0]
mtu_request=9216
/usr/share/openvswitch/scripts/ovs-ctl restart
```

6. Start the VM and configure the MTU on the VM.

```
ifconfig eth0 11.100.124.2/16 up
ifconfig eth0 mtu 9216
ping 11.100.126.1 -s 9188 -M do -c1
```

2.3.6.5.2.7 E2E Cache

This feature is at beta level.

OVS offload rules are based on a multi-table architecture. E2E cache feature enables merging the multi-table flow matches and actions into one joint flow.

This improves connection tracking performance by using a single-table when exact match is detected.

➤ To set the E2E cache size (default = 4k):

```
ovs-vsctl set open_vswitch . other_config:e2e-size=<size>
systemctl restart openvswitch
```

Note: Make sure to restart the openvswitch service in order for the configuration to take effect.

➤ To enable/disable E2E cache (default = disabled) :

```
ovs-vsctl set open_vswitch . other_config:e2e-enable=<true/false>
systemctl restart openvswitch
```

Note: Make sure to restart the openvswitch service in order for the configuration to take effect.

➤ To run E2E cache statistics:

```
ovs-appctl dpctl/dump-e2e-stats
```

➤ To run E2E cache flows:

```
ovs-appctl dpctl/dump-e2e-flows
```

2.3.6.5.2.8 Geneve Encapsulation/Decapsulation

Geneve tunneling offload feature support includes matching on extension header.

➤ To configure OVS-DPDK Geneve encap/decap:

1. Create a br-phy bridge.

```
ovs-vsctl --may-exist add-br br-phy -- set Bridge br-phy datapath_type=netdev -- br-set-external-id br-phy
bridge-id br-phy -- set bridge br-phy fail-mode=standalone
```

2. Attach PF interface to br-phy bridge.

```
ovs-vsctl add-port br-phy pf -- set Interface pf type=dppk options:dppk-devargs=<PF PCI>
```

3. Configure IP to the bridge.

```
ifconfig br-phy <$local_ip_1> up
```

4. Create a br-int bridge.

```
ovs-vsctl --may-exist add-br br-int -- set Bridge br-int datapath_type=netdev -- br-set-external-id br-int
bridge-id br-int -- set bridge br-int fail-mode=standalone
```

5. Attach representor to br-int.

```
ovs-vsctl add-port br-int rep$x -- set Interface rep$x type=dppk options:dppk-devargs=<PF
PCI>,representor=[$x]
```

6. Add a port for the GENEVE tunnel.

```
ovs-vsctl add-port br-int geneve0 -- set interface geneve0 type=geneve options:key=<VNI>
options:remote_ip=<$remote_ip_1> options:local_ip=<$local_ip_1>
```

2.3.6.5.2.9 Parallel Offloads

OVS-DPDK supports parallel insertion and deletion of offloads (flow & CT). While multiple threads are supported, by default only one is used.

➤ To configure multiple threads:

```
ovs-vsctl set Open_vSwitch . other_config:n-offload-threads=3
```

Make sure to restart the openvswitch service in order for the configuration to take effect.

```
systemctl restart openvswitch
```

For more information, see the [OvS user manual](#).

2.3.6.5.2.10 sFlow

This feature allows for monitoring traffic sent between two VMs on the same host using an sFlow collector.

To sample all traffic over the OVS bridge, run the following:

```
# ovs-vsctl -- --id=@sflow create sflow agent="\${SFLOW_AGENT}" \  
target="\${SFLOW_TARGET}:${SFLOW_HEADER}" header=${SFLOW_HEADER} \  
sampling=${SFLOW_SAMPLING} polling=10 \  
-- set bridge sflow=@sflow
```

Parameter	Description
SFLOW_AGENT	Indicates that the sFlow agent should send traffic from SFLOW_AGENT's IP address
SFLOW_TARGET	Remote IP address of the sFLOW collector
SFLOW_PORT	Remote IP destination port of the sFlow collector
SFLOW_HEADER	Size of packet header to sample (in bytes)
SFLOW_SAMPLING	Sample rate

To clear the sFLOW configuration, run the following:

```
# ovs-vsctl clear bridge br-vxlan mirrors
```

Currently sFlow for OVS-DPDK is supported without CT.

2.3.6.5.2.11 CT CT NAT

To enable ct-ct-nat offloads in OvS-DPDK, execute the following command (default value is false):

```
ovs-vsctl set open_vswitch . other_config:ct-action-on-nat-conns=true
```

If disabled, ct-ct-nat configurations will not be fully offloaded, improving connection offloading rate for other cases (ct and ct-nat).

If enabled, ct-ct-nat configurations will be fully offloaded but ct and ct-nat offloading will be slower to be created.

2.3.6.5.2.12 OpenFlow Meters (OpenFlow13+):

OpenFlow meters in OVS are implemented according to RFC 2697 (Single Rate Three Color Marker–srTCM).

- The srTCM meters an IP packet stream and marks its packets either green, yellow, or red. The color is decided on a Committed Information Rate (CIR) and two associated burst sizes, Committed Burst Size (CBS), and Excess Burst Size (EBS).
- A packet is marked green if it does not exceed the CBS, yellow if it exceeds the CBS but not the EBS, and red otherwise.
- The volume of green packets should never be smaller than the CIR.

To configure a meter in OVS:

1. Create a meter over a certain bridge:

a.

```
ovs-ofctl -O openflow13 add-meter $bridge
meter=$id,$pktps/$kbps,band=type=drop,rate=$rate,[burst,burst_size=$burst_size]
```

- b. Parameters:

Parameter	Description
bridge	Name of the bridge on which the meter will be applied.
id	Unique meter ID (32 bits) which will be used as an identifier for the meter.
pktps/kbps	Indication if the meter should work according to packets-per-second or kilobits-per-second.
rate	Rate of pktps/kbps of allowed data transmission.
burst	If set, enables burst support for meter bands through the “burst_size” parameter.
burst_size	If burst is specified for the meter entry, configures the maximum burst allowed for the band in kilobits/packets, depending on whether kbps or pktps was specified. If unspecified, the switch is free to select some reasonable value depending on its configuration. Currently, if burst was not specified, the burst_size parameter is set as the “rate”.

2. Add the meter to a certain OpenFlow rule. For example:

```
ovs-ofctl -O openflow13 add-flow $bridge "table=0,actions=meter:$id,normal"
```

3. View the meter statistics:

```
ovs-ofctl -O openflow13 meter-stats $bridge meter=$id
```

4. For more information, refer to openvswitch documentation <http://www.openvswitch.org/support/dist-docs/ovs-ofctl.8.txt>

2.3.6.6 VirtIO Acceleration through Hardware vDPA

2.3.6.6.1 Hardware vDPA Installation

Hardware vDPA requires QEMU v2.12 (or with upstream 6.1.0) and DPDK v20.11 as minimal versions.

➤ *To install QEMU:*

1. Clone the sources:

```
git clone https://git.qemu.org/git/qemu.git
cd qemu
git checkout v2.12
```

2. Build QEMU:

```
mkdir bin
cd bin
../configure --target-list=x86_64-softmmu --enable-kvm
make -j24
```

➤ *To install DPDK:*

1. Clone the sources:

```
git clone git://dpdk.org/dpdk
cd dpdk
git checkout v20.11
```

2. Install dependencies (if needed):

```
yum install cmake gcc libnl3-devel libudev-devel make pkgconfig valgrind-devel pandoc libibverbs libmlx5
libmnl-devel -y
```

3. Configure DPDK:

```
export RTE_SDK=$PWD
make config T=x86_64-native-linuxapp-gcc
cd build
sed -i 's/(CONFIG_RTE_LIBRTE_MLX5_PMD=)n/ly/g' .config
sed -i 's/(CONFIG_RTE_LIBRTE_MLX5_VDPA_PMD=)n/ly/g' .config
```

4. Build DPDK:

```
make -j
```

5. Build the vDPA application:

```
cd $RTE_SDK/examples/vdpa/
make -j
```

2.3.6.6.2 Hardware vDPA Configuration

➤ *To configure huge pages:*

```
mkdir -p /hugepages
mount -t hugetlbfs hugetlbfs /hugepages
echo <more> > /sys/devices/system/node/node0/hugepages/hugepages-1048576kB/nr_hugepages
echo <more> > /sys/devices/system/node/node1/hugepages/hugepages-1048576kB/nr_hugepages
```

➤ **To configure a vDPA VirtIO interface in an existing VM's xml file (using libvirt):**

1. Open the VM's configuration xml for editing:

```
virsh edit <domain name>
```

2. Modify/add the following:

- a. Change the top line to:

```
<domain type='kvm' xmlns:qemu='http://libvirt.org/schemas/domain/qemu/1.0'>
```

- b. Assign a memory amount and use 1GB page size for hugepages (size must be the same as used for the vDPA application), so that the memory configuration looks like the following.

```
<memory unit='KiB'>4194304</memory>
<currentMemory unit='KiB'>4194304</currentMemory>
<memoryBacking>
  <hugepages>
    <page size='1048576' unit='KiB' />
  </hugepages>
</memoryBacking>
```

- c. Assign an amount of CPUs for the VM CPU configuration, so that the `vcpu` and `cputune` configuration looks like the following.

```
<vcpu placement='static'>5</vcpu>
<cputune>
  <vcpupin vcpu='0' cpuset='14' />
  <vcpupin vcpu='1' cpuset='16' />
  <vcpupin vcpu='2' cpuset='18' />
  <vcpupin vcpu='3' cpuset='20' />
  <vcpupin vcpu='4' cpuset='22' />
</cputune>
```

- d. Set the memory access for the CPUs to be shared, so that the `cpu` configuration looks like the following.

```
<cpu mode='custom' match='exact' check='partial'>
  <model fallback='allow'>Skylake-Server-IBRS</model>
  <numa>
    <<cell id='0' cpus='0-4' memory='8388608' unit='KiB' memAccess='shared' />
  </numa>
</cpu>
```

- e. Set the emulator in use to be the one built in step "[2. Build QEMU](#)" above, so that the emulator configuration looks as follows.

```
<emulator><path to qemu executable></emulator>
```

- f. Add a virtio interface using qemu command line argument entries, so that the new interface snippet looks as follows.

```
<qemu:commandline>
  <qemu:arg value='-chardev' />
  <qemu:arg value='socket,id=charnet1,path=/tmp/sock-virtio0' />
  <qemu:arg value='-netdev' />
  <qemu:arg value='vhost-user,chardev=charnet1,queues=16,id=hostnet1' />
  <qemu:arg value='-device' />
```

```
<qemu:arg value='virtio-net-pci,mq=on,vectors=6,netdev=hostnet1,id=net1,mac=e4:11:c6:d3:45:f2,bus
=pci.0,addr=0x6,
page-per-vq=on,rx_queue_size=1024,tx_queue_size=1024' />
</qemu:commandline>
```

Note: In this snippet, the vhostuser socket file path, the amount of queues, the MAC and the PCI slot of the VirtIO device can be configured.

2.3.6.6.3 Running Hardware vDPA

Hardware vDPA supports SwitchDev mode only.

➤ *Create the ASAP² environment:*

1. Create the VFs.
2. Enter switchdev mode.
3. Set up OVS.

➤ *Run the vDPA application.*

```
cd $RTE_SDK/examples/vdpa/build
./vdpa -w <VF PCI BDF>,class=vdpa --log-level=pmd,info -- -i
```

➤ *Create a vDPA port via the vDPA application CLI.*

```
create /tmp/sock-virtio0 <PCI DEVICE BDF>
```

Note: The vhostuser socket file path must be the one used when configuring the VM.

➤ *Start the VM.*

```
virsh start <domain name>
```

For further information on the vDPA application, please visit: https://doc.dpdk.org/guides/sample_app_ug/vdpa.html.

2.3.6.7 Bridge Offload

- Bridge offload is supported on ConnectX-6 Dx NIC
- Bridge offload is supported SwitchDev mode only
- Bridge offload is supported from kernel version 5.15

A Linux bridge is in-kernel software network switch (based on and implements subset of IEEE 802.1D standard) that is used to connect Ethernet segments together in a protocol-independent way. Packets are forwarded based on L2 Ethernet header addresses.

mlx5 provides capabilities to offload bridge data-plane unicast packet forwarding and VLAN management to hardware.

2.3.6.7.1 Basic Configuration

1. Initialize the ASAP² environment:
 - a. Create the VFs.
 - b. Enter switchdev mode.
2. Create a bridge and add mlx5 representors to bridge:

```
ip link add name bridge0 type bridge
ip link set enp8s0f0_0 master bridge0
```

2.3.6.7.2 Configuring VLAN

1. Enable VLAN filtering on the bridge.

```
ip link set bridge0 type bridge vlan_filtering 1
```

2. Configure port VLAN matching (trunk mode). In this configuration, only packets with specified VID are allowed.

```
bridge vlan add dev enp8s0f0_0 vid 2
```

3. Configure port VLAN tagging (access mode). In this configuration VLAN header is pushed/popped on reception/transmission on port.

```
bridge vlan add dev enp8s0f0_0 vid 2 pvid untagged
```

2.3.6.7.3 VF LAG Support

Bridge supports offloading on bond net device that is fully initialized with mlx5 uplink representors and is in single (shared) FDB LAG mode. Details about initialization of LAG are provided in [SR-IOV VF LAG](#) section, above.

Add bonding net device to bridge.

```
ip link set bond0 master bridge0
```

For further information on interacting with Linux bridge via iproute2 bridge tool, please consult man page (man 8 bridge).

2.3.6.8 Appendix: NVIDIA Firmware Tools

Download and install the MFT package corresponding to your computer's operating system. You would need the kernel-devel or kernel-headers RPM before the tools are built and installed. The package is available at [nvidia.com/en-us/networking/](https://www.nvidia.com/en-us/networking/) → Products → Software → Firmware Tools.

1. Start the mst driver.

```
# mst start
Starting MST (Mellanox Software Tools) driver set
```

```
Loading MST PCI module - Success
Loading MST PCI configuration module - Success
Create devices
```

2. Show the devices status.

```
ST modules:
-----
MST PCI module loaded
MST PCI configuration module loaded

PCI devices:
-----
DEVICE_TYPE          MST          PCI          RDMA NET          NUMA
ConnectX4lx(rev:0)  /dev/mst/mt4117_pciconf0.1  04:00.1  net-enp4s0f1  NA
ConnectX4lx(rev:0)  /dev/mst/mt4117_pciconf0    04:00.0  net-enp4s0f0  NA

# mlxconfig -d /dev/mst/mt4117_pciconf0 q | head -16

Device #1:
-----

Device type:      ConnectX4lx
PCI device:       /dev/mst/mt4117_pciconf0

Configurations:          Current
SRIOV_EN                 True(1)
NUM_OF_VFS               8
PF_LOG_BAR_SIZE         5
VF_LOG_BAR_SIZE         5
NUM_PF_MSIX              63
NUM_VF_MSIX              11
LINK_TYPE_P1             ETH(2)
LINK_TYPE_P2             ETH(2)
```

3. Make sure your configuration is as follows:

- * SR-IOV is enabled (SRIOV_EN=1)
 - * The number of enabled VFs is enough for your environment (NUM_OF_VFS=N)
 - * The port's link type is Ethernet (LINK_TYPE_P1/2=2) when applicable
- If this is not the case, use `mlxconfig` to enable that, as follows:

- a. Enable SR-IOV.

```
# mlxconfig -d /dev/mst/mt4115_pciconf0 s SRIOV_EN=1
```

- b. Set the number of required VFs.

```
# mlxconfig -d /dev/mst/mt4115_pciconf0 s NUM_OF_VFS=8
```

- c. Set the link type to Ethernet.

```
# mlxconfig -d /dev/mst/mt4115_pciconf0 s LINK_TYPE_P1=2
# mlxconfig -d /dev/mst/mt4115_pciconf0 s LINK_TYPE_P2=2
```

4. Conduct a cold reboot (or a firmware reset).

```
# mlxfwreset -d /dev/mst/mt4115_pciconf0 reset
```

5. Query the firmware to make sure everything is set correctly.

```
# mlxconfig -d /dev/mst/mt4115_pciconf0 q
```

2.4 Troubleshooting

You may be able to easily resolve the issues described in this section. If a problem persists and you are unable to resolve it yourself, please contact your NVIDIA representative or NVIDIA Support at networking-support@nvidia.com.

The chapter contains the following sections:

- [General Issues](#)
- [Ethernet Related Issues](#)
- [Installation Related Issues](#)
- [Performance Related Issues](#)
- [SR-IOV Related Issues](#)
- [OVS Offload Using ASAP2 Direct Related Issues](#)

2.4.1 General Issues

Issue	Cause	Solution
The system panics when it is booted with a failed adapter installed.	Malfunction hardware component	<ol style="list-style-type: none"> 1. Remove the failed adapter. 2. Reboot the system.
NVIDIA adapter is not identified as a PCI device.	PCI slot or adapter PCI connector dysfunctionality	<ol style="list-style-type: none"> 1. Run <code>lspci</code>. 2. Reseat the adapter in its PCI slot or insert the adapter to a different PCI slot. If the PCI slot confirmed to be functional, the adapter should be replaced.
NVIDIA adapters are not installed in the system.	Misidentification of the NVIDIA adapter installed	<p>Run the command below and check NVIDIA's MAC to identify the NVIDIA adapter installed.</p> <pre>lspci grep Mellanox' or 'lspci -d 15b3:</pre> <p>Note: NVIDIA MACs start with: 00:02:C9:xx:xx:xx, 00:25:8B:xx:xx:xx or F4:52:14:xx:xx:xx"</p>
Insufficient memory to be used by udev upon OS boot.	udev is designed to fork() new process for each event it receives so it could handle many events in parallel, and each udev instance consumes some RAM memory.	Limit the udev instances running simultaneously per boot by adding <code>udev.children-max=<number></code> to the kernel command line in grub.
Operating system running from root file system located on a remote storage (over NVIDIA devices), hang during reboot/shutdown (errors such as "No such file or directory" will appear).	The <code>mlnx-en.d</code> service script is called using the 'stop' option by the operating system. This option unloads the driver stack. Therefore, the OS root file system disappears before the reboot/shutdown procedure is completed, leaving the OS in a hang state.	Disable the <code>openibd</code> 'stop' option by setting 'ALLOW_STOP=no' in <code>/etc/mlnx-en.conf</code> configuration file.

2.4.2 Ethernet Related Issues

Issue	Cause	Solution
Ethernet interfaces renaming fails leaving them with names such as renameXY.	Invalid udev rules.	<p>Review the udev rules inside the "/etc/udev/rules.d/70-persistent-net.rules" file. Modify the rules such that every rule is unique to the target interface, by adding correct unique attribute values to each interface, such as dev_id, dev_port and KERNELS or address).</p> <p>Example of valid udev rules:</p> <pre>SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?* ", ATTR{dev_id}=="0x0", ATTR{type} =="1", KERNEL=="eth*", ATTR{dev_port}=="0", KER- NELS=="0000:08:00.0", NAME="eth4" SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?* ", ATTR{dev_id}=="0x0", ATTR{type} =="1", KERNEL=="eth*", ATTR{dev_port}=="1", KER- NELS=="0000:08:00.0", NAME="eth5"</pre>
No link.	Misconfiguration of the switch port or using a cable not supporting link rate.	<ul style="list-style-type: none"> • Ensure the switch port is not down • Ensure the switch port rate is configured to the same rate as the adapter's port
Degraded performance is measured when having a mixed rate environment (10GbE, 40GbE and 56GbE).	Sending traffic from a node with a higher rate to a node with lower rate.	<p>Enable Flow Control on both switch ports and nodes:</p> <ul style="list-style-type: none"> • On the server side run: <pre>ethtool -A <interface> rx on tx on</pre> • On the switch side run the following command on the relevant interface: <pre>send on force and receive on force</pre>
No link with break-out cable.	Misuse of the break-out cable or misconfiguration of the switch's split ports	<ul style="list-style-type: none"> • Use supported ports on the switch with proper configuration. For further information, please refer to the MLNX_OS User Manual. • Make sure the QSFP breakout cable side is connected to the SwitchX.

2.4.3 Installation Related Issues

2.4.3.1 Application Binary Interface (ABI) Incompatibility with MLNX_EN Kernel Modules

This section is relevant for RedHat and SLES distributions only.

2.4.3.1.1 Overview

MLNX_EN package for RedHat comes with RPMs that support KMP (weak-modules), meaning that when a new errata kernel is installed, compatibility links will be created under the weak-updates directory for the new kernel. Those links allow using the existing MLNX_EN kernel modules without the need for recompilation. However, at times, the ABI of the new kernel may not be compatible with the MLNX_EN modules, which will prevent loading them. In this case, the MLNX_EN modules must be rebuilt against the new kernel.

2.4.3.1.2 Detecting ABI Incompatibility with MLNX_EN Modules

When MLNX_EN modules are not compatible with a new kernel from a new OS or errata kernel, no links will be created under the weak-updates directory for the new kernel, causing the driver load to fail. Checking for the existence of needed module links under weak-updates directory can be done by reloading the MLNX_EN modules. If one or more modules are missing, the driver reload will fail with an error message.

Example:

```
*****
# /etc/init.d/mlnx-en.d restart
Unloading HCA driver:                [ OK ]
Loading HCA driver and Access Layer: [ OK ]
Module rdma_cm belong to kernel which is not a part of MLNX[FAILED]ipping...
Loading rdma_ucm                      [FAILED]
*****
```

2.4.3.1.2.1 Resolving ABI Incompatibility with MLNX_EN Modules

In order to fix ABI incompatibility with MLNX_EN modules, the modules should be recompiled against the new kernel, using the `mlnx_add_kernel_support.sh` script, available in MLNX_EN installation image.

There are two ways to recompile the MLNX_EN modules:

1. Local recompilation and installation on one server.

Run the `install` command to recompile the kernel modules and reinstall the whole MLNX_EN on the server. Mount MLNX_EN ISO image or extract the TGZ file:

```
# cd <MLNX_EN dir>
# ./install --skip-distro-check --add-kernel-support --kmp --force
```

Notes:

- The `--kmp` flag will enable rebuilding RPMs with KMP (weak-updates) support for the new kernel. Therefore, in the next OS/kernel update, the same modules can be used with the new kernel (assuming that the ABI compatibility was not broken again).
- The command above will rebuild only the kernel RPMs (using `mlnx_add_kernel_support.sh`), and will save the resulting MLNX_EN package under `/tmp` and start installing it automatically. This package can be used for installation on other servers using regular `install` command or `yum`.

2. Preparing a new image on one server and deploying it on the cluster.

- a. Use the `mlnx_add_kernel_support.sh` script directly only to rebuild the kernel RPMs (without running any installations) on one server. Mount MLNX_EN ISO image or extract the TGZ file:

```
# cd <MLNX_EN dir>
# ./mlnx_add_kernel_support.sh -m $PWD --kmp -y
```

Note: This command will save the resulting MLNX_EN package under `/tmp`.

Example:

```
*****
# cd /tmp/MLNX_EN_LINUX-5.2-2.2.0.0-DB-rhel7.8-x86_64
# ./mlnx_add_kernel_support.sh -m $PWD --kmp -y
Note: This program will create mlnx-en TGZ for rhel7.8 under /tmp directory.
See log file /tmp/mlnx_iso.28286_logs/mlnx_ofed_iso.28286.log

Checking if all needed packages are installed...
Building mlnx-en RPMs . Please wait...

Creating metadata-rpms for 3.10.0-1127.el7.x86_64 ...
WARNING: If you are going to configure this package as a repository, then please note
WARNING: that it contains unsigned rpms, therefore, you need to disable the gpgcheck
WARNING: by setting 'gpgcheck=0' in the repository conf file.
Created /tmp/mlnx-en-5.3-1.0.0.1-rhel7.8-x86_64-ext.tgz
*****
```

- b. Install the newly created MLNX_EN package on the cluster:

Option 1: Copy the package to the servers and install it using the `install` script.

Option 2: Deploy the MLNX_EN package using YUM (for YUM installation instructions, refer to [Installing MLNX_EN Using YUM](#) section):

- i. Extract the resulting MLNX_EN image and copy it to a shared NFS location.
- ii. Create a YUM repository configuration.
- iii. Install the new MLNX_EN kernel RPMs on the servers: `# yum update`

Example:

```
*****
...
...
=====
Package      Arch      Version
Repository  Size
=====
Updating:
epel-release    noarch    7-7
epel            14 k
kmod-iser      x86_64    1.8.0-OFED.3.3.1.0.0.1.gf583963.201606210906.rhel7u1
mlnx_ofed      35 k
kmod-iser      x86_64    1.0-OFED.3.3.1.0.0.1.gf583963.201606210906.rhel7u1
mlnx_ofed      32 k
kmod-kernel-mft-mlnx x86_64    4.4.0-1.201606210906.rhel7u1
mlnx_ofed      10 k
kmod-knem-mlnx x86_64    1.1.2.90mlnx1-OFED.3.3.0.0.1.0.3.1.ga04469b.201606210906.rhel7u1
mlnx_ofed      22 k
kmod-mlnx-ofa_kernel x86_64    3.3-OFED.3.3.1.0.0.1.gf583963.201606210906.rhel7u1
mlnx_ofed      1.4 M
kmod-srp       x86_64    1.6.0-OFED.3.3.1.0.0.1.gf583963.201606210906.rhel7u1
mlnx_ofed      39 k
*****
```

```
Transaction Summary
=====
Upgrade 7 Packages
...
*****
```

Note: The MLNX_EN user-space packages will not change; only the kernel RPMs will be updated. However, “YUM update” can also update other inbox packages (not related to OFED). In order to install the MLNX_EN kernel RPMs only, make sure to run:

```
# yum install mlnx-en-kernel-only
```

Note: `mlnx-en-kernel-only` is a metadata RPM that requires the MLNX_EN kernel RPMs only.

- c. Verify that the driver can be reloaded:

```
# /etc/init.d/mlnx-en.d restart
```

2.4.4 Performance Related Issues

Issue	Cause	Solution
The driver works but the transmit and/or receive data rates are not optimal.	-	<p>These recommendations may assist with gaining immediate improvement:</p> <ol style="list-style-type: none"> 1. Confirm PCI link negotiated uses its maximum capability 2. Stop the IRQ Balancer service: <code>/etc/init.d/irq_balancer stop</code> 3. Start <code>mlnx_affinity</code> service: <code>mlnx_affinity start</code> <p>For best performance practices, please refer to the "Performance Tuning Guide for NVIDIA Network Adapters".</p>
Out of the box throughput performance in Ubuntu14.04 is not optimal and may achieve results below the line rate in 40GE link speed.	IRQ affinity is not set properly by the <code>irq_balancer</code>	For additional performance tuning, please refer to Performance Tuning Guide.

2.4.5 SR-IOV Related Issues

Issue	Cause	Solution
When assigning a VF to a VM the following message is reported on the screen: <code>PCI-assgine: error: requires KVM support</code>	SR-IOV and virtualization are not enabled in the BIOS.	<ol style="list-style-type: none"> 1. Verify they are both enabled in the BIOS 2. Add to the GRUB configuration file to the following kernel parameter: <code>"intel_immun=on"</code> (see "Setting Up SR-IOV" section).

2.4.6 OVS Offload Using ASAP2 Direct Related Issues

Issue	Cause	Solution(s)
Traffic is not offloaded	<p>OVS uses TC flower classifier to add offloading rules to both the software and the hardware.</p> <ul style="list-style-type: none"> TC flower classifier fails to add a rule. A rule was added to the TC flower classifier but failed to be added to the firmware. 	<ul style="list-style-type: none"> Check for system error in dmesg or the system logging facility like journalctl Check OVS logs for errors Dump the rules using the TC command line For example: Dump rules on a specific interface <pre># tc filter show dev ens4f0 parent ffff:</pre>

2.5 Common Abbreviations and Related Documents

Common Abbreviations and Acronyms

Abbreviation/Acronym	Description
B	(Capital) 'B' is used to indicate size in bytes or multiples of bytes (e.g., 1KB = 1024 bytes, and 1MB = 1048576 bytes)
b	(Small) 'b' is used to indicate size in bits or multiples of bits (e.g., 1Kb = 1024 bits)
FW	Firmware
HCA	Host Channel Adapter
HW	Hardware
IB	InfiniBand
iSER	iSCSI RDMA Protocol
LSB	Least significant <i>byte</i>
lsb	Least significant <i>bit</i>
MSB	Most significant <i>byte</i>
msb	Most significant <i>bit</i>
NIC	Network Interface Card
SW	Software
VPI	Virtual Protocol Interconnect
IPoIB	IP over InfiniBand
PFC	Priority Flow Control
PR	Path Record
RoCE	RDMA over Converged Ethernet
SL	Service Level
SRP	SCSI RDMA Protocol
MPI	Message Passing Interface

Abbreviation/Acronym	Description
QoS	Quality of Service
ULP	Upper Layer Protocol
VL	Virtual Lane
vHBA	Virtual SCSI Host Bus Adapter
uDAPL	User Direct Access Programming Library

Glossary

The following is a list of concepts and terms related to InfiniBand in general and to Subnet Managers in particular. It is included here for ease of reference, but the main reference remains the *InfiniBand Architecture Specification*.

Term	Description
Channel Adapter (CA), Host Channel Adapter (HCA)	An IB device that terminates an IB link and executes transport functions. This may be an HCA (Host CA) or a TCA (Target CA)
HCA Card	A network adapter card based on an InfiniBand channel adapter device
IB Devices	An integrated circuit implementing InfiniBand compliant communication
IB Cluster/Fabric/Subnet	A set of IB devices connected by IB cables
In-Band	A term assigned to administration activities traversing the IB connectivity only
Local Identifier (ID)	An address assigned to a port (data sink or source point) by the Subnet Manager, unique within the subnet, used for directing packets within the subnet
Local Device/Node/System	The IB Host Channel Adapter (HCA) Card installed on the machine running IBDIAG tools
Local Port	The IB port of the HCA through which IBDIAG tools connect to the IB fabric
Master Subnet Manager	The Subnet Manager that is authoritative, that has the reference configuration information for the subnet
Multicast Forwarding Tables	A table that exists in every switch providing the list of ports to forward received multicast packet. The table is organized by MLID
Network Interface Card (NIC)	A network adapter card that plugs into the PCI Express slot and provides one or more ports to an Ethernet network
Standby Subnet Manager	A Subnet Manager that is currently quiescent, and not in the role of a Master Subnet Manager, by the agency of the master SM
Subnet Administrator (SA)	An application (normally part of the Subnet Manager) that implements the interface for querying and manipulating subnet management data
Subnet Manager (SM)	One of several entities involved in the configuration and control of the IB fabric
Unicast Linear Forwarding Tables (LFT)	A table that exists in every switch providing the port through which packets should be sent to each LID
Virtual Protocol Interconnect (VPI)	An NVIDIA technology that allows NVIDIA channel adapter devices (ConnectX®) to simultaneously connect to an InfiniBand subnet and a 10GigE subnet (each subnet connects to one of the adapter ports)

Related Documentation

Document Name	Description
InfiniBand Architecture Specification, Vol. 1, Release 1.2.1	The InfiniBand Architecture Specification that is provided by IBTA
IEEE Std 802.3ae™-2002 (Amendment to IEEE Std 802.3-2002) Document # PDF: SS94996	Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications Amendment: Media Access Control (MAC) Parameters, Physical Layers, and Management Parameters for 10 Gb/s Operation
Firmware Release Notes for NVIDIA adapter devices	See the Release Notes relevant to your adapter device
MFT User Manual and Release Notes	NVIDIA Firmware Tools (MFT) User Manual and Release Notes documents
WinOF User Manual	Mellanox WinOF User Manual describes the installation, configuration, and operation of NVIDIA Windows driver
VMA User Manual	NVIDIA VMA User Manual describes the installation, configuration, and operation of NVIDIA VMA driver

3 Documentation History

- [Release Notes History](#)
- [User Manual Revision History](#)

3.1 Release Notes History

- [Release Notes Change Log History](#)
- [Bug Fixes History](#)

3.1.1 Release Notes Change Log History

This section includes history of changes and new feature of three major (GA) releases back. For older versions' history, please refer to their dedicated release notes.

Supported Cards	Description
All HCAs	Supported in the following adapter cards <u>unless specifically stated otherwise:</u> ConnectX-4 / ConnectX -4 Lx / ConnectX-5 / ConnectX-6 / ConnectX-6 Dx / ConnectX-6 Lx / ConnectX-7 / BlueField-2
ConnectX-6 Dx and above	Supported in the following adapter cards <u>unless specifically stated otherwise:</u> ConnectX-6 Dx / ConnectX-6 Lx / ConnectX-7 / BlueField-2
ConnectX-6 and above	Supported in the following adapter cards <u>unless specifically stated otherwise:</u> ConnectX-6 / ConnectX-6 Dx / ConnectX-6 Lx / ConnectX-7 / BlueField-2
ConnectX-5 and above	Supported in the following adapter cards <u>unless specifically stated otherwise:</u> ConnectX-5 / ConnectX-6 / ConnectX-6 Dx / ConnectX-6 Lx / ConnectX-7 / BlueField-2
ConnectX-4 and above	Supported in the following adapter cards <u>unless specifically stated otherwise:</u> ConnectX-4 / ConnectX -4 Lx / ConnectX-5 / ConnectX-6 / ConnectX-6 Dx / ConnectX-6 Lx / ConnectX-7 / BlueField-2

Feature/Change	Description
5.8-4.1.5.0	
Operating Systems	Added support for the following operating systems: <ul style="list-style-type: none"> • RHEL 9.3 • RHEL 8.9 • Debian 10.13
Bug Fixes	Bug Fixes in This Version
Feature/Change	Description
5.8-3.0.7.0	

Operating Systems	Added support for the following operating systems: <ul style="list-style-type: none"> • RHEL 9.2 • RHEL 8.8 • SLES15-SP5 • KYLIN10-SP3
General	Bug fixes
5.8-2.0.3.0	
General	Bug fixes
5.8-1.1.2.1	
General	Bug fixes
5.8-1.0.1.1	
Remove Dependency Between SR-IOV and eSwitch Mode	[All HCAs] Removed dependency between SR-IOV and eSwitch mode. Currently, there are three eSwitch modes: none, legacy, and switchdev (non of which are the default mode). When disabling SR-IOV, the current eSwitch mode will be changed to none. This feature removes eSwitch mode none and also removes dependency between SR-IOV and eSwitch mode.
DevLink Parallel Command	[All HCAs] Added support for running DevLink commands in parallel on different DevLink devices is possible. For example, burning firmware on a few cards on the same host in parallel using DevLink API is now possible.
Graceful Shutdown of Parent and Page Supplier	[All HCAs] Set default graceful period values for functions based on their type. ECPFs will get graceful period of 3 minutes, PFs get 1 minute, and VFs/SFs get 30 seconds.
N Pulses Per Second (NPPS)	[ConnectX-6 Dx and above] Enhanced NPPS to allow setting a pulse period higher than 1 pulse per second and to allow setting the pulse width. If the width is unset, the driver implicitly sets it to half the given period (the width should be less than the pulse period). In this release, the pulse duration ranges between 65536 NS-524288 NS.
Remote Invalidate Option for MKeys	[All HCAs] Added support for the option to enable remote invalidation when creating a new mkey. This way the rkey for a memory region can be changed frequently.
GPUDirect Over DMA-BUF	[All HCAs] Added support for GPUDirect support over dma-buf. As such, using the new mechanism <code>nv_peer_mem</code> is no longer required. The following is required for dma-buf support: <ul style="list-style-type: none"> • Linux kernel version 5.12 or later • OpenRM version 515 or later Perftest support was added as well: Default option in perftest is without dmabuf. To run with this option, add <code>--use_cuda_dmabuf</code> in addition to <code>use_cuda</code> flag.
General	Bug fixes

Feature/ Change	Description
5.7-1.0.2.0	
Support Representor Metering Over SFs	[ConnectX-6 Dx and above and BlueField-2] Extended the support of representor metering from supporting only VFs representor to also supporting SFs representor.
Exposing Error Counters on a VPort Manager	[ConnectX-4 and above] Added support for exposing error counters on a VPort manager function for all other VPorts. These counters can be used to detect malicious users who are exploiting flows that can slow the device. The counters are exposed through debugfs under: /sys/kernel/debug/mlx5/esw/<func>/vnic_diag/
Memory Consumption Minimization	[ConnectX-4 and above] Added support for providing knobs which enable users to minimize memory consumption of mlx5 functions (PF/VF/SF).
XDP Support for Uplink Representors	[ConnectX-5 and ConnectX-6 Dx) Added XDP support for uplink representors in switchdev mode.
Resiliency to tx_port_ts	[ConnectX-6 Dx and above] Added resiliency to the tx_port_ts feature. private-flag may be enabled via ethtool tx_port_ts which provides a more accurate time-stamp. In very rare cases, the said time-stamp was lost, leading to losing the synchronization altogether. This feature allows for fast recovery and allows to quickly regain synchronization.
Database of Devlink Health Asserts	[ConnectX-4 and above] Health buffer now contains more debug information like the epoch time in sec of the error and the error's severity. The print to dmesg is done with the debug level corresponding to the error's severity. This allows the user to use dmesg attribute: dmesg --level to focus on different severity levels of firmware errors.
Expose FEC Counters via Ethtool	[ConnectX-5 and above] Exposed the following FEC (forward error detection) counters: ETHTOOL_A_FEC_STAT_CORRECTED <ul style="list-style-type: none"> • fc_fec_corrected_blocks_laneX • rs_fec_corrected_blocks ETHTOOL_A_FEC_STAT_UNCORR <ul style="list-style-type: none"> • fc_fec_uncorrectable_blocks_laneX • rs_fec_uncorrectable_blocks ETHTOOL_A_FEC_STAT_CORR_BITS <ul style="list-style-type: none"> • phy_corrected_bits Command: ethtool -I show-fec <ifc>
Application Device Queues	[ConnectX-4 and above] Added driver-level support for Application Device Queues. This feature allows partition defining over the RX/TX queues into groups and isolates traffic of different applications. This mainly improves predictability and tail latency.
Reinjection of Packets Into Kernel	[All HCAs] Added support for a new software steering action, mlx5dv_dr_action_create_dest_root_table(). This action can be used to forward packets back into a level 0 table. As a table with level 0 is the kernel owned table, this will result in injecting packets to the kernel steering pipeline.

Feature/Change	Description
5.7-1.0.2.0	
DCT LAG	[ConnectX-6 Dx and above] Added firmware support to allow explicit port selection based on steering and not QP affinity. Functionality: 1. Use LAG Hash Mode for the HCA with two ports, if supported. 2. Keep port affinity function in LAG Hash Mode if it supports bypass select flow table in non-SwitchDev mode.
AES-XTS in RDMA	Added support for plaintext AES-XTS DEKs.
General	Bug fixes

3.1.1.1 Customer Affecting Changes

Customer Affecting Change	Description
5.9-0.5.6.0	
Deprecation, LAG Mode via Sysfs	Setting LAG mode via Sysfs is going to be deprecated in a future release. Instead, LAG Hash mode will be used by default, similar to upstream behavior.
LAG Configuration, PCI Error	From version 5.9, LAG configuration will be lost in case driver incurs a PCI error. Make sure to reconfigure the bond after driver completes the recovery from the PCI error. In releases prior to 5.9, in case of PCI error (EEH injections on PPC setup), the driver recovers LAG bond and reconfigures it automatically in case it what configured before the appearance of the error.

Customer Affecting Change	Description
5.7-1.0.2.0	
Multi-Block Encryption	Multi-block encryption is currently unsupported, due to a hardware limitation.

Feature/Change	Description
5.6-2.0.9.0	
Operating Systems	Added support for the following Operating Systems: RHEL8.6, RHEL9.0, SLES15-SP4.
General	Bug fixes

Feature/Change	Description
5.6-1.0.3.5	
General	
New Adapter Card Support	Added support for ConnectX-7 adapter cards. ConnectX-7 has the same feature set as the ConnectX-6 adapter card.
ASAP² Features	
Bridge Spoof Check	[All HCAs] Added support for spoof check with TC flower rules on representors attached to bridge to mirror spoof check SR-IOV functionality.
Setting VF Group Rate Limit	[ConnectX-5 and above] Added support for setting VF group rate limit using Devlink command.
TC Flows on Shared Block	[ConnectX-5 and above] Added support for creation of TC flows on shared block of VF representors.
Flow Metering	[ConnectX-6 Dx and above] Added support for offloading OpenFlow Meters in OVS-DPDK. Please note the following: <ul style="list-style-type: none"> • Meter offload can be applied only on port 0 and it's VFs • Only one meter per flow is allowed • Only one meter band per meter is allowed • Only meter band type drop is supported • Meter-stats might not be accurate
Core Features	
Firmware Reset	[BlueField-2] Added support of firmware reset in DPU NIC mode.
Increased Robustness of mlx5_core Driver Recovery	[All HCAs] Increased the firmware pre-initialization timeout from 2 minutes to 2 hours when waiting for firmware during driver health recovery, allowing the driver to passively recover from a firmware reset, even if the reset takes an unusually long time. Additionally, added an exit clause to the wait for firmware loop, allowing immediate response to a user initiated device removal.
NetDev Features	
Ethtool CQE Mode Control	[ConnectX-4 and above] Replaced the vendor-specific Ethtool API (priv-flag) with a standard Ethtool API (replaced 'ethtool --set-priv-flags ethX rx_cqe_moder on/off tx_cqe_moder on/off' with 'ethtool -C ethX cqe-mode-rx on/off cqe-mode-tx on/off'). This decreases the amount of vendor-specific configurations and aligns mlx5 driver with the upstream Ethtool API.
SyncE	[ConnectX-6 Dx] Added an indication in SyncE Daemon that states whether SyncE engine moved to holdover state due to failure (the reason for failure will be displayed). In addition, added indication whether SyncE engines collected enough frequency samples in order to move to holdover.
Security	
OVS-IPSec Full Offload	[BlueField-2] Added support for configuration of IPsec full offload using OVS by adding VXLAN tunnel to OVS with the PSK option.
Installation	

Installation	<p>New options were added to the ofed_uninstall.sh script: <code>--only-kernel</code> and <code>--only-user</code>. Those can be used to uninstall only kernel packages or only user-space packages (the equivalent of kernel-only install or user-space-only install, respectively). This may be useful to keep different sets of kernel and user-space installations.</p> <p>When running the uninstall script with a combination of <code>--only-kernel</code> and <code>--only-user</code> produced an undefined result.</p>
--------------	--

Feature/Change	Description
5.5-1.0.3.2	
ASAP² Features	
Bridge Offloads with VLAN	[ConnectX-4 and above] Added support for bridge offloads with VLAN support that works on top of mlx5 representors in switchdev mode.
Supporting OVS Groups in Fast-Failover Mode	[ConnectX-6 Dx] Improved OVS failover through support for OVS groups in fast-failover mode + VF_LAG configuration with OVS.
Exposing Hairpin Queues Information	<p>[ConnectX-6 Dx and BlueField-2] Added support for exposing hairpin out of buffer drop counter per device. This feature shows buffer drops related only to hairpin queues which were opened on the queried device.</p> <p>To enable this counting mode (this must be done before any hairpin rules are created), use the following: <code>echo "on <peer_devname>" > /sys/class/net/<dev>/hp_oob_cnt_mode</code> where <peer_devname> is the peer device to which traffic coming to the configured device will be forwarded to for transmission.</p> <p>To read the drop counter, use the following: <code>cat /sys/class/net/<dev>/hp_oob_cnt</code></p>
Linux Bridge Offload	[ConnectX-6 Dx and BlueField-2] Added bridge offloads to support bonding (VF LAG), attaching bond device to bridge instead of uplink representors.
VLAN Pop/Push	[ConnectX-6 Dx] Added OOB support for VLAN push on Rx (wire to VF) and VLAN pop on Tx (wire to VF) in switchdev mode.
Offload Forwarding to Multiple Destinations	[ConnectX-5 and above] Added support for offloading packet replication to up to 32 destination through the use of TC rule.
Slow Path Metering	<p>[ConnectX-4 and above] Expanding the RDMA statistic tool to support setting vendor-specific optional counters dynamically using netlink.</p> <p>Added to mlx5_ib the following optional counters: <code>cc_rx_ce_pkts,cc_rx_cnp_pkts,cc_tx_cnp_pkts</code>.</p> <p>Example: <code>\$ rdma statistic mode supported link rocep8s0f0/1</code> <code>link rocep8s0f0/1 supported optional-counters</code> <code>cc_rx_ce_pkts,cc_rx_cnp_pkts,cc_tx_cnp_pkts</code> <code>\$ sudo rdma statistic set link rocep8s0f0/1 optional-counters</code> <code>cc_rx_ce_pkts,cc_rx_cnp_pkts</code> <code>\$ rdma statistic mode link rocep8s0f0/1</code> <code>link rocep8s0f0/1 optional-counters cc_rx_ce_pkts,cc_rx_cnp_pkts</code> <code>\$ sudo rdma statistic set link rocep8s0f0/1 optional-counters cc_rx_ce_pkts</code> <code>\$ rdma statistic mode link rocep8s0f0/1</code> <code>link rocep8s0f0/1 optional-counters cc_rx_ce_pkts</code> <code>\$ sudo rdma statistic unset link rocep8s0f0/1 optional-counters</code></p>
Core Features	

Subfunction Trust Configuration Enhancement	[ConnectX-5 and above] Added support via mlxdevm to mark a given PCI subfunction (SF) or virtual function (VF) as a trusted function. The device/firmware decides how to define privileges and access to resources.
Prevent VF Memory Exhaustion	[All] Added support for preventing VF memory exhaustion. This feature exposes a sysfs (to the system admin) which can set a limit on each VF memory consumption. Note: Currently only supported on Ethernet.
BlueField NIC Separate Reset	[BlueField-2] Added support for resetting the NIC domain of BlueField-2 while keeping ARM alive.
Multiple Steering Priorities for FDB Rules	[ConnectX-6 Dx and BlueField-2] Added support in multiple flow steering priorities for FDB rules.
NetDev Features	
Traffic Engineering: Hierarchical QoS	[ConnectX-5 and above] Added support for offloading the HTB qdisc to the NIC, allowing it to scale better by eliminating a single locking point. The configuration is done with the TC commands. Note: Kernel 5.15 or higher is required. Limited to 256 nodes.
TLS RX Resynchronization Resiliency Feature Description	[ConnectX-6 Dx and above] Added support for driver resiliency against high load of RX resync operations.
Simultaneous PTP and CQE Compression	Added support for the activation of PTP and CQE compression simultaneously. Since CQE compression might harm the accuracy of the PTP, the feature enables PTP packets to be moved to a dedicated queue where they are not subjected to compression. However, this configuration conflicts with setting aRFS. Turning off CQE compression, causes a hiccup in traffic which may cause a loss of synchronization. To overcome this, restart the synchronization. Note: This combination is supported only for Ethernet drivers. Other driver profiles, like IPoB and representors, do not support this combination.
Installation Features	
Multiple Development Headers Packages	Allowed installing multiple mlnx-ofa_kernel development headers packages (for different kernel versions of the same mlnx-ofa_kernel package version) side by side on the same system.
Kernel Module Signature	Added signature of kernel modules of EulerOS 2.0 SP8-SP10 (x86_64 and aarch64) builds of MLNX_OFED.
Enable sf-cfg-drv by Default in EulerOS2.0	Enabled SF_CFG (SF config dummy driver, --with-sf-cfg-drv) on EulerOS2.0 SP8 and SP10.

Feature/ Change	Description
5.4-1.0.3.0	
ASAP²	
Enlarge Switchdev Tables	[ConnectX-5 and above] Added support for allowing OVS kernel to support up to 128 matches (groups) per table and 16M entries per group.

Offloading Extended ct_state Flags	<p>[ConnectX-5 and above] Added support to offload ct_state flags rpl, inv, and rel.</p> <ul style="list-style-type: none"> For rpl, support was added for both set and not set matching offload (i.e., +rpl and -rpl). For inv and rel, support was added only for the not set option (i.e., -rel and -inv).
Core	
Auxiliary Bus in mlx5 Driver	[ConnectX-4] Updated mlx5 driver to use auxiliary bus in order to integrate different driver components into driver core and optimize module load/unload sequences.
Installation	
Script Removal from mlnx-ofa_kernel	[General] Moved all Python scripts and some other common scripts out of the mlnx-ofa_kernel packages. This removed the python dependency from that package when rebuilding it and avoided unnecessary errors when rebuilding them for custom kernels.
Netdev	
What-Just-Happened (WJH) in NICs	<p>[ConnectX-4] Added support for WJH in NICs. WJH allows for visibility of dropped packets (i.e., receiving notice of drop counters increase, seeing content of the dropped packets, debugging, and more).</p> <p>WJH is a service in devlink context and it is already implemented in the switch.</p> <p>Note: processing dropped packets (even for visibility purposes) may cause a degradation in performance and leaves the driver vulnerable for malicious attacks. The feature is disabled by default.</p> <p>Supported traps:</p> <ul style="list-style-type: none"> VLAN mismatch: existing generic trap DEVLINK_TRAP_GENERIC_ID_DMACH_MISMATCH Traps received packets with wrong VLAN tag DMAC mismatch: new generic trap DEVLINK_TRAP_GENERIC_ID_DMACH_MISMATCH Traps received packets with wrong destination MAC <p>Support added in user-space (N/A or package name + version): Devlink infrastructure (man7.org/linux/man-pages/man8/devlink-trap.8.html)</p> <p>Devlink provides an infrastructure called devlink trap which allow a device to register/unregister and to enable/disable traps. Devlink traps also provide traps grouping and policing. The trapped packets are monitored and then forward to the drop monitor. Drop monitor is used to send notifications to user space about dropped packets.Note: For this release, NIC WJH will not implement the policy.</p>
ethtool Extended Link State	<p>[General] Added ethtool extended link state to mlx5e.</p> <p>ethtool can be used to get more information to help troubleshoot the state. For example, if there is no link due to missing cable, run the following:</p> <pre>\$ ethtool eth1 ... Link detected: no (No cable)</pre> <p>Besides the general extended state, drivers can pass additional information about the link state using the sub-state field.</p> <p>Example:</p> <pre>\$ ethtool eth1 ... Link detected: no (Autoneg, No partner detected)</pre> <p>The extended state is available only for some cases of no link. In other cases, ethtool will print only "Link detected: no" as it did before.</p>
RDMA	

DV "Signature API"	[ConnectX-5 and above] Added support for "Signature API" which, on supported devices, allows application-level data-integrity checks via a signature handover mechanism. Various signature types, including CRC32 and T10-DIF, can be automatically calculated and checked, stripped, or appended during the transfer at full wire speed.
ibv_query_qp_data_in_order() verb	[General] Added support for ibv_query_qp_data_in_order() API. This API enables an application to check if the given QP data is guaranteed to be in order, enabling poll for data instead of poll for completion.
Relaxed Ordering for Kernel ULPs	[ConnectX-4] Added support for enabling Relaxed Ordering for Kernel ULPs. Using relaxed ordering can improve performance in some setups. Since kernel ULPs are expected to support RO, it is enabled for them by default so they can benefit from it.
ah_to_qp Mapping	[ConnectX-6 Dx] Added support for mapping a QP to AH over DEVX API, which enables DC/UD QPs to use multiple CC algorithms in the same data center.
Steering UserSpace	
Matching on RAW Tunnel Headers	[ConnectX-5 and above] Added DR support for matching on RAW tunnel headers using the misc5 parameters, This feature allows matching on each bit of the header, including reserved fields.
Software Steering Insertion Rate Optimizations	[ConnectX-6 Dx] Added support for better insertion rate in software steering. This includes multi-QP which skips areas in the code that may be for debug only.
Software Steering Rule Optimization	[ConnectX-6 Dx] Improved rate of updating steering rules, insertion, and deletion. The feature includes definers, multi-qp approach, and better memory usage.
Duplicate Rules Insertion	[ConnectX-5 and above] Added support for ability to allow or prevent insertion of duplicate rules, so the user can choose one of the following behaviors: 1. Prevent duplicate rules, so that already-existing rule and fail can be detected. 2. Allow duplicate rules, to enable updating the rule's action (this will only take effect once the previous rule is deleted). By default, duplicate rules are allowed.
Improved Software Steering Rule Creation Stability	[ConnectX-6 Dx] Made it so that all rule's insertion occur in a defined time using defined (export) size of Htbl and decreased use of dynamic allocation.

Feature/ Change	Description
MLNX_EN 5.2-1.0.4.0	
Rx Multi-strides CQE Compression	[ConnectX-5 and above] Added CQE compression support for Rx multi-strides packets.
Multi-application QoS	[ConnectX-5 and above] Added support for configuring QoS on a single QP or on a group of QPs.
MPLS-over-UDP Hardware Offload Support	[ConnectX-5 and above] Added support for encap/decap hardware offload of IPv4 traffic over MPLS-over-UDP. This can be used in networks with MPLS routers to achieve more efficient routing.
Connection Tracking with Hairpin	[ConnectX-5 and above] Added support for adding connection tracking rules on VFs to forward traffic from one VF to the other.

Feature/ Change	Description
MLNX_EN 5.2-1.0.4.0	
sFlow Sampling Rules Offload	[ConnectX-5 and above] Added support for offloading sFlow sampling rules. sFlow is an industry standard technology for monitoring high speed switched networks. Open vSwitch integrated sFlow to extend the visibility into virtual servers, ensuring data center visibility and control. Added support for offloading sFlow sampling rules.
mlx5dv_dr Software Steering Parallel Rules Insertion	[ConnectX-5 & ConnectX-6 Dx/ BlueField & BlueField-2] Added support for a locking mechanism to enable parallel insertion of rules into the software steering using the mlx5dv_dr API. The parallel insertion improves the insertion rate and takes place when adding Rx and Tx rules via the FDB domain.
mlx5dv_dr API Matching on Geneve Tunnel	[ConnectX-5 & ConnectX-6 Dx/ BlueField & BlueField-2] Added support for the option to match mlx5dv_dr API on Geneve tunnel using a dynamic flex parser. The option header consists of class, type, length and data. The parser should be configured using devx command, after which a rule can be created to match on parser ID and data.
OVS-DPDK Geneve Encap/ Decap	[ConnectX-5 & ConnectX-6 Dx/ BlueField & BlueField-2] Added support for Geneve tunneling offload, including matching on extension header.
OVS-DPDK Parallel Offloads	[ConnectX-5 & ConnectX-6 Dx/ BlueField & BlueField-2] Added support for parallel insertion and deletion of offloaded rules using multiple OVS threads.
GTP-U TEID Modification	[BlueField-2 & ConnectX-6 Dx] [Beta] Added support to modify GTP-U TEID. This support requires flex parser configuration.
OVS-DPDK E2E Cache Support	[BlueField-2 & ConnectX-6 Dx] [Beta] Improved performance of OVS Connection Tracking flows by enabling the merge of the multi-table flow matches and actions into one joint flow.
Tx Port Time-Stamping	[ConnectX-6 Dx and above] Transmitted packet timestamping accuracy can be improved when using a timestamp generated at the port level instead of a timestamp generated upon CQE creation. Tx port time-stamping better reflects the actual time of a packet's transmission. This feature is disabled by default. The feature can be enabled or disabled using the following command. <code>ethtool --set-priv-flags <ifs-name> tx_port_ts on / off</code> For further information on this feature, please see Tx Port Time-Stamping .
Tunnel Rules Offload	[ConnectX-6 Dx and above] Added support for offloading tunnel rules when the source interface is VF (in addition to uplink) in the Hypervisor.
	[ConnectX-6 Dx and above] Added support for offloading tunnel rules when the source interface is OpenvSwitch bridge (internal port).
Connection Tracking Mirroring Offload	[ConnectX-6 Dx and above] Added support for using Mirroring Offload with Connection Tracking.
mlx5dv_dr API ASO Flow Meter	[ConnectX-6 Dx and above] Added support for ASO flow meter using the mlx5dv_dr API, which allows for monitoring the packet rate for specific flows. When a packet hits a flow that is connected to a flow meter, the rate of packets through this meter is evaluated, and the packet is marked with a color copied into one of the C registers, according to the current rate compared to the reference rate.

Feature/ Change	Description
MLNX_EN 5.2-1.0.4.0	
mlx5dv_dr API ASO First Hit	[ConnectX-6 Dx and above] Added support for ASO first hit using the mlx5dv_dr API, which allows for tracking rule hits by packets. When a packet hits a rule with the ASO first hit action, a flag is set indicating this event, and the original value of the flag is copied to one of the C registers.
mlx5dv_dr API GTP-U Extension Header	[ConnectX-6 Dx and above] Added mlx5dv_dr API support for matching on a new field "gtpu_first_ext_dw_0". This field enables packet filtering based on the GTP-U first extension header (first dword only). To enable parsing of tunnel GTP-U extension header, run the following command. <pre>./cloud_fw_reset.py FLEX_PARSER_PROFILE_ENABLE=3</pre>
IPsec Offload	[ConnectX-6 Lx and above] Added IPsec full offload support for extended sequence number, replay protection window and lifetime packet limit.
Firmware Upgrade	[All HCAs] Firmware upgrade during MLNX_EN installation is now done on all supported devices simultaneously rather than consecutively.
RDMA-CM Disassociate Support	[All HCAs] Added support for connecting kernel and RDMA-CM in a reliable way based on device index.
New Query GID API	[All HCAs] Added support for a new query GID API that allows for querying a single GID entry by its port and GID index, or querying for all GID tables of a specific device. This API works over ioctl instead of sysfs, which accelerates the querying process.
Multi-Host Firmware Reset	[All HCAs] Added support for performing multi-host firmware reset in order to upgrade the device firmware. Firmware reset loads the new firmware in case it was burnt on the flash and was pending activation, and reloads the current firmware image from the flash in case no new firmware was pending.
Firmware Live Patching	[All HCAs] [Alpha] Added support for firmware live patching in the driver. Live patching updates the firmware without the need to perform firmware reset. However, it can only be applied in scenarios where the difference between the current and new firmware versions are minor, which is decided upon by the firmware itself.
Devlink Firmware Reset	[All HCAs] Added support in the devlink tool for performing firmware reset in order to upgrade the device firmware. Firmware reset loads the new firmware in case it was burnt on the flash and was pending activation, and reloads the current firmware image from the flash in case no new firmware was pending. For further information, please refer to the the devlink man page. Note: In order for the firmware reset to run successfully, the following conditions should be met. <ul style="list-style-type: none"> • Each function should have the driver up and active with a version that supports this feature • None of the functions has the devlink parameter <code>enable_remote_dev_reset</code> set to False. <div style="border: 1px solid orange; padding: 10px; margin-top: 10px;"> <p>The current MLNX_EN does not include the latest iproute2 version that provides support for this feature. Therefore, to be able to work with it, make sure to install the latest iproute2 version available on Github.</p> </div>
Command Interface Resiliency	[All HCAs] Added a resiliency mechanism for the driver to manually poll the command event queue (EQ) in case of a command timeout. In case the resiliency mechanism finds unhandled event queue entry (EQE) due to a lost interrupt, the driver will handle it, after which the command interface returns to a healthy state.

Feature/ Change	Description
MLNX_EN 5.2-1.0.4.0	
Offloaded Traffic Sniffer	[All HCAs] Setting a sniffer private flag is deprecated and no longer required. In order to capture offloaded/RoCE traffic, tcpdump can now be run on the RDMA device.
Devlink Port Health Reporters	[All HCAs] Added per-port reporters to devlink health to manage per-port health activities. Users can now access the devlink port reporters by specifying the port index in addition to the device devlink name through the devlink health commands API. This update was first introduced in iproute2 v5.8. As part of this feature, mlx5e Tx and Rx reporters are now redefined as devlink port reporters. For examples, please see devlink-health manpage.
Memory Registration Optimization	[All HCAs] Optimized memory consumption of memory registration in huge page systems. As an example, in a 2MB huge page system, 600 MB would be saved for 100 GB memory registration.
mlx5dv API	[All HCAs] Added support for mlx5dv API to modify the configured UDP source port for RoCE packets of a given RC/UC QP when QP is in RTS state.
Bug Fixes	See Bug Fixes .
Innova IPsec NIC Support	[Innova IPsec] Removed support for the network adapter Innova IPsec (EN).

Category	Description
Rev 5.1-1.0.4.0	
IP-in-IP RSS Offload	[ConnectX-4 and above] Added support for receive side scaling (RSS) offload in IP-in-IP (IPv4 and IPv6).
Devlink Port Support in Non-representor Mode	[ConnectX-4 and above] Added support for viewing the mlx5e physical devlink ports using the 'devlink port' command. This also may affect network interface names, if predictable naming scheme is configured. Suffix indicating a port number will be added to interface name.
Devlink Health State Notifications	[ConnectX-4 and above] Added support for receiving notifications on devlink health state changes when an error is reported or recovered by one of the reporters. These notifications can be seen using the userspace 'devlink monitor' command.
Legacy SR-IOV VF LAG Load Balancing	[ConnectX-4 and above] When VF LAG is in use, round-robin the Tx affinity of channels among the different ports, if supported by the firmware, enables all SQs of a channel to share the same port affinity. This allows the distribution of traffic sent from a VF between two ports, as well as round-robin the starting port among VFs to distribute traffic originating from single-core VMs.
RDMA-CM DevX Support	[ConnectX-4 and above] Added support for DevX in RDMA-CM applications.
RoCEv2 Flow Label and UDP Source Port Definition	[ConnectX-4 and above] This feature provides flow label and UDP source port definition in RoCE v2. Those fields are used to create entropy for network routes (ECMP), load balancers and 802.3ad link aggregation switching that are not aware of RoCE headers.
RDMA Tx Steering	[ConnectX-4 and above] Enabled RDMA Tx steering flow table. Rules in this flow table will allow for steering transmitted RDMA traffic.
Custom Parent-Domain Allocators for CQ	[ConnectX-4 and above] Enabled specific custom allocations for CQs.

Category	Description
Rev 5.1-1.0.4.0	
mlx5dv Helper APIs for Tx Affinity Port Selection	[ConnectX-4 and above] Added support for the following mlx5dv helper APIs which enable the user application to query or set a RAW QP's Tx affinity port number in a LAG configuration. <ul style="list-style-type: none"> • <code>mlx5dv_query_qp_lag_port</code> • <code>mlx5dv_modify_qp_lag_port</code>
RDMA-CM Path Alignment	[ConnectX-4 and above] Added support for RoCE network path alignment between RDMA-CM message and QP data. The drivers and network components in RoCE calculate the same hash results for egress port selection both on the NICs and the switches.
CQ and QP Context Exposure	[ConnectX-4 and above] Exposed QP, CQ and MR context in raw format via RDMA tool.
In-Driver xmit_more	[ConnectX-4 and above] Enabled <code>xmit_more</code> feature by default in kernels that lack Rx bulking support (v4.19 and above) to ensure optimized IP forwarding performance when stress from Rx to Tx flow is insufficient. In kernels with Rx bulking support, <code>xmit_more</code> is disabled in the driver by default, but can be enabled to achieve enhanced IP forwarding performance.
Relaxed Ordering	[ConnectX-4 and above] Relaxed ordering is a PCIe feature which allows flexibility in the transaction order over the PCIe. This reduces the number of retransmissions on the lane, and increases performance up to 4 times. By default, mlx5e buffers are created with Relaxed Ordering support when firmware capabilities are on and the PCI subsystem reports that CPU is not on the kernel's blocklist. Note: Some CPUs which are not listed in the kernel's blocklist may suffer from buggy implementation of relaxed ordering, in which case the user may experience a degradation in performance and even unexpected behavior. To turn off relaxed ordering and restore previous behavior, run <code>setpci</code> command as instructed here . Example: <code>"RlxdOrd-" : setpci -s82:00.0 CAP_EXP+8.w=294e</code>
ODP Huge Pages Support	[ConnectX-4 and above] Enabled ODP Memory Region (MR) to work with huge pages by exposing <code>IBV_ACCESS_HUGETLB</code> access flag to indicate that the MR range is mapped by huge pages. The flag is applicable only in conjunction with <code>IBV_ACCESS_ON_DEMAND</code> .
Offloaded Traffic Sniffer	[ConnectX-4 and above] Removed support for Offloaded Traffic Sniffer feature and replaced its function with Upstream solution <code>tcpdump</code> tool.
Connection Tracking Offload	[ConnectX-5 and above] Added support for offloading TC filters containing connection tracking matches and actions.
Dual-Port RoCE Support	[ConnectX-5 and above] Enabled simultaneous operation of dual-port RoCE and Ethernet in SwitchDev mode.
IP-in-IP Tunnel Offload for Checksum and TSO	[ConnectX-5 and above] Added support for the driver to offload checksum and TSO in IP-in-IP tunnels.
Packet Pacing DevX Support	[ConnectX-5 and above] Enabled RiverMax to work over DevX with packet pacing functionality by exposing a few DV APIs from <code>rdma-core</code> to enable allocating/destroying a packet pacing index. For further details on usage, see man page for: <code>mlx5dv_pp_alloc()</code> and <code>mlx5dv_pp_free()</code> .
Software Steering Support for Memory Reclaiming	[ConnectX-5 and above] Added support for reclaiming device memory to the system when it is not in use. This feature is disabled by default and can be enabled using the command <code>mlx5dv_dr_domain_set_reclaim_device_memory()</code> .

Category	Description
Rev 5.1-1.0.4.0	
SR-IOV Live Migration	[ConnectX-5 and above] [Beta] Added support for performing a live migration for a VM with an SR-IOV NIC VF attached to it and with minimal to no traffic disruption. This feature is supported in SwitchDev mode; enabling users to fully leverage VF TC/OVS offloads, where the failover inbox driver is in the Guest VM, and the bonding driver is in the Hypervisor. Note that you must use the latest QEMU and libvirt from the Upstream github.com sources.
Uplink Represor Modes	[ConnectX-5 and above] Removed support for new_netdev mode in SwitchDev mode. The new default behaviour is to always keep the NIC netdev.
OVS-DPDK Offload Statistics	[ConnectX-5 and above] Added support for dumping connection tracking offloaded statistics.
OVS-DPDK Connection Tracking Labels Exact Matching	[ConnectX-5 and above] Added support for labels exact matching in OVS-DPDK CT openflow rules.
Kernel Software Managed Flow Steering (SMFS) Performance	[ConnectX-5 and above] Improved the performance of Kernel software steering by reducing its memory consumption.
OVS-DPDK LAG Support	[ConnectX-5 & ConnectX-6 Dx] Added support for LAG (modes 1,2,4) with OVS-DPDK.
Get FEC Status on PAM4/50G	[ConnectX-6 and above] Allowed configuration of Reed Solomon and Low Latency Reed Solomon over PAM4 link modes.
RDMA-CM Enhanced Connection Establishment (ECE)	[ConnectX-6 and above] Added support for allowing automatic enabling/disabling of vendor specific features during connection establishment between network nodes, which is performed over RDMA-CM messaging interface.
RoCE Selective Repeat	[ConnectX-6 and above] This feature introduces a new QP retransmission mode in RoCE in which dropped packet recovery is done by re-sending the packet instead of re-sending the PSN window only (Go-Back-N protocol). This feature is enabled by default when RDMA-CM is being used and both connection nodes support it.
IPsec Full Offload	[ConnectX-6 Dx & BlueField-2] [Beta] Added support for IPsec full offload (VxLAN over ESP transport).
Hardware vDPA on OVS-DPDK	[ConnectX-6 Dx & BlueField-2] Added support for configuring hardware vDPA on OVS-DPDK. This support includes the option to fall back to Software vDPA in case the NIC installed on the driver does not support hardware vDPA.
IPsec Crypto Offloads	[ConnectX-6 Dx] Support for IPsec Crypto Offloads feature over ConnectX-6 Dx devices and up is now at GA level.
TLS Tx Hardware Offload	[ConnectX-6 Dx] Support for TLS Tx Hardware Offload feature over ConnectX-6 Dx devices and up is now at GA level.
TLS Rx Hardware Offload	[ConnectX-6 Dx] [Alpha] Added support for hardware offload decryption of TLS Rx traffic over crypto-enabled ConnectX-6 Dx NICs and above.

Category	Description														
Rev 5.1-1.0.4.0															
Userspace Software Steering ConnectX-6 Dx Support	[ConnectX-6 Dx] Support for software steering on ConnectX-6 Dx adapter cards in the user-space RDMA-Core library through the mlx5dv_dr API is now at GA level.														
Kernel Software Steering ConnectX-6 Dx Support	[ConnectX-6 Dx] [Beta] Added support for kernel software steering on ConnectX-6 Dx adapter cards.														
Adapters	[ConnectX-6 Lx] Added support for ConnectX-6 Lx adapter cards.														
RDMA-Core Migration	[All HCAs] As of MLNX_EN v5.1, Legacy verbs libraries have been fully replaced by RDMA-Core library. For the list of new APIs used for various MLNX_EN features, please refer to the Migration to RDMA-Core document .														
Firmware Reactivation	[All HCAs] Added support for safely inserting consecutive firmware images without the need to reset the NIC in between.														
UCX-CUDA Support	[All HCAs] UCX-CUDA is now supported on the following OSs and platforms. <table border="1" style="width: 100%; margin-top: 10px;"> <thead> <tr> <th style="text-align: center;">OS</th> <th style="text-align: center;">Platform</th> </tr> </thead> <tbody> <tr> <td>RedHat 7.6 ALT</td> <td>PPC64LE</td> </tr> <tr> <td>RedHat 7.7</td> <td>x86_64</td> </tr> <tr> <td>RedHat 7.8</td> <td>PPC64LE/x86_64</td> </tr> <tr> <td>RedHat 7.9</td> <td>x86_64</td> </tr> <tr> <td>RedHat 8.1</td> <td>x86_64</td> </tr> <tr> <td>RedHat 8.2</td> <td>x86_64</td> </tr> </tbody> </table>	OS	Platform	RedHat 7.6 ALT	PPC64LE	RedHat 7.7	x86_64	RedHat 7.8	PPC64LE/x86_64	RedHat 7.9	x86_64	RedHat 8.1	x86_64	RedHat 8.2	x86_64
OS	Platform														
RedHat 7.6 ALT	PPC64LE														
RedHat 7.7	x86_64														
RedHat 7.8	PPC64LE/x86_64														
RedHat 7.9	x86_64														
RedHat 8.1	x86_64														
RedHat 8.2	x86_64														
HCOLL-CUDA	[All HCAs] The hcoll package includes a CUDA plugin (hmca_gpu_cuda.so). As of MLNX_EN v5.1, it is built on various platforms as the package hcoll-cuda. It will be installed by default if the system has CUDA 10-2 installed. Notes: <ul style="list-style-type: none"> If you install MLNX_EN from a package repository, you will need to install the package hcoll-cuda explicitly to be able to use it. HCOLL-CUDA is supported on the same OSs that include support for UCX-CUDA (listed in the table above), except for RedHat 8.1 and 8.2. 														
GPUDirect Storage (GDS)	[All HCAs] [Beta] Added support for the new technology of GDS (GPUDirect Storage) which enables a direct data path between local or remote storage, such as NFS, NVMe or NVMe over Fabric (NVMe-oF), and GPU memory. Both GPUDirect RDMA and GPUDirect Storage avoid extra copies through a bounce buffer in the CPU's memory. They enable the direct memory access (DMA) engine near the NIC or storage to move data on a direct path into or out of GPU memory, without burdening the CPU or GPU. To enable the feature, run <code>./install --with-nfsrdma --with-nvmf --enable-gds --add-kernel-support</code> To get access to GDS Beta, please reach out to the GDS team at GPUDirectStorageExt@nvidia.com . For the list of operating systems on which GDS is supported, see here .														
Bug Fixes	See Bug Fixes .														

Category	Description
Rev 5.0-1.0.0.0	
Adapters	[ConnectX-6 Dx] Added support for ConnectX-6 Dx adapter cards.
Userspace Software Steering ConnectX-6 Dx Support	[Beta] Added support for software steering on ConnectX-6 Dx adapter cards in the userspace RDMA-Core library through the mlx5dv_dr API.
Virtual Output Queuing (VoQ) Counters	[ConnectX-6 Dx and above] Exposed rx_prio[p]_buf_discard, rx_prio[p]_wred_discard and rx_prio[p]_marked firmware counters that count the number of packets that were dropped due to insufficient resources.
IPsec Crypto Offloads	[ConnectX-6 Dx and above] [Beta] IPsec crypto offloads are now supported on ConnectX-6 Dx devices and up. The offload functions use the existing ip xfrm tool to activate offloads on the device. It supports transport/tunnel mode with AES-GCM IPsec scheme.
TLS TX Hardware Offload	[ConnectX-6 Dx and above, not including ConnectX-6 Lx] [Alpha] Added support for hardware offload encryption of TLS traffic.
VirtIO Acceleration through Datapath I/O Processor (vDPA)	[ConnectX-6 Dx and above] Added support to enable mapping the VirtIO access region (VAR) to be used for doorbells by vDPA applications. Specifically, the following DV APIs were introduced (see man page for more details): <ul style="list-style-type: none"> • mlx5dv_alloc_var() • mlx5dv_free_var()
Resource Allocation on External Memory	[ConnectX-5 and above] Added support to enable overriding mlx5 internal allocations in order to let applications allocate some resources on external memory, such as that of the GPU. The above is achieved by extending the parent domain object with custom allocation callbacks. Currently supported verbs objects are: QP, DBR, RWQ, SRQ.
Hardware Clock Exposure	[ConnectX-5 and above] Added support for querying the adapter clock via mlx5dv_query_device.
ODP Diagnostic Counters	[ConnectX-5 and above] Added ODP diagnostics counters for the following items per MR (memory region) within IB/mlx5 driver: <ol style="list-style-type: none"> 1. Page faults: Total number of faulted pages. 2. Page invalidations: Total number of pages invalidated by the OS during all invalidation events. The translations can no longer be valid due to either non-present pages or mapping changes. 3. Prefetched pages: When prefetching a page, a page fault is generated in order to bring the page to the main memory.
Devlink Health CR-Space Dump	[ConnectX-5 and above] Added the option to dump configuration space via the devlink tool in order to improve debug capabilities.
Multi-packet TX WQE Support for XDP Transmit Flows	[ConnectX-5 and above] The conventional TX descriptor (WQE or Work Queue Element) describes a single packet for transmission. Added driver support for the HW feature of multi-packet TX WQEs in XDP transmit flows. With this, the HW becomes capable of working with a new and improved WQE layout that describes several packets. In effect, this feature saves PCI bandwidth and transactions, and improves transmit packet rate.
OVS-Kernel ToS Rewrite	[ConnectX-5 and above] Added support for Type of Service (ToS) rewrite in the OVS-Kernel.
OVS-Kernel Mirroring	[ConnectX-5 and above] Added support for mirroring output in SwitchDev mode in the OVS-Kernel. The mirroring port may either be a local or a remote VF, using VxLAN or GRE encapsulations.

Category	Description
Rev 5.0-1.0.0.0	
GENEVE Encap/Decap Rules Offload	[ConnectX-5 and above] Added support for GENEVE encapsulation/decapsulation rules offload.
GPRS Tunneling Protocol (GTP) Header	[ConnectX-5 and above] [Beta] Added support for matching (filtering) GTP header-based packets using mlx5dv_dr API over user-space RDMA-Core library.
Multi Packet Tx WQE Support for XDP Transmit Flows	[ConnectX-5 and above] Added driver support for the hardware feature of multi-packet Tx to work with a new and improved WQE layout that describes several packets instead of a single packet for XDP transmission flows. This saves PCI bandwidth and transactions, and improves transmit packet rate.
Userspace Software Steering Debugging API	[ConnectX-5 and above] [Beta] Added support for software steering to dump flows for debugging purposes in the user-space RDMA-Core library through the mlx5dv_dr API.
Kernel Software Steering for Connection Tracking (CT)	[ConnectX-5 and above] [Beta] Added support for updating CT rules using the software steering mechanism.
Kernel Software Steering Remote Mirroring	[ConnectX-5 and above] [Beta] Added support for updating remote mirroring rules using the software steering mechanism.
Discard Counters	[ConnectX-4 and above] Exposed rx_prio[p]_discards discard counters per priority that count the number of received packets dropped due to lack of buffers on the physical port.
MPLS Traffic	[ConnectX-4 and above] Added support for reporting TSO and CSUM offload capabilities for MPLS tagged traffic and, allowed the kernel stack to use these offloads.
mlx5e Max Combined Channels	[ConnectX-4 and above] Increased the driver's maximal combined channels value from 64 to 128 (however, note that OOB value will not cross 64). 128 is the upper bound. Lower maximal value can be seen on the host, depending on the number of cores and MSIX's configured by the firmware.
RoCE Accelerator Counters	[ConnectX-4 and above] Added the following RoCE accelerator counters: <ul style="list-style-type: none"> • roce_adp_retrans - counts the number of adaptive retransmissions for RoCE traffic • roce_adp_retrans_to - counts the number of times RoCE traffic reached timeout due to adaptive retransmission • roce_slow_restart - counts the number of times RoCE slow restart was used • roce_slow_restart_cnps - counts the number of times RoCE slow restart generated CNP packets • roce_slow_restart_trans - counts the number of times RoCE slow restart changed state to slow restart
Migration to RDMA-Core	[All HCAs] The default installation of the userspace is now the RDMA-Core library instead of the legacy verbs. This achieves most of the legacy experimental verbs' functionalities, and more. For NVIDIA VMA or NVIDIA RiverMax, use experimental verbs (prefix "ibv_exp"). For further information on the migration to RDMA-Core and the list of new APIs used for various MLNX_EN features, please refer to the Migration to RDMA-Core document .
ibdev2netdev Tool Output	[All HCAs] ibdev2netdev tool output was changed such that the bonding device now points at the bond instead of the slave interface.
Memory Region	[All HCAs] Added support for the user to register memory regions with a relaxed ordering access flag. This can enhance performance, depending on architecture and scenario.

Category	Description
Rev 5.0-1.0.0.0	
Devlink Health Reporters	[All HCAs] Added support for monitoring and recovering from errors that occur on the RX queue, such as cookie errors and timeout.
GSO Optimization	[All HCAs] Improved GSO (Generic Segmentation Offload) workload performance by decreasing doorbells usage to the minimum required.
TX CQE Compression	[All HCAs] Added support for TX CQE (Completion Queue Element) compression. Saves on outgoing PCIe bandwidth by compressing CQEs together. Disabled by default. Configurable via private flags of ethtool.
Firmware Versions Query via Devlink	[All HCAs] Added the option to query for running and stored firmware versions using the devlink tool.
Firmware Flash Update via Devlink	[All HCAs] Added the option to update the firmware image in the flash using the devlink tool. Usage: <code>devlink dev flash <dev> file <file_name>.mfa2</code> For further information on how to perform this update, see "Updating Firmware Using ethtool/devlink and .mfa2 File" section in MFT User Manual.
Devlink Health WQE Dump	[All HCAs] Added support for WQE (Work Queue Element) dump, triggered by an error on Rx/Tx reporters. In addition, some dumps (not triggered by an error) can be retrieved by the user via devlink health reporters.
GENEVE Tunnel Stateless Offload	[All HCAs] Added support for GENEVE tunneled hardware offloads of TSO, CSUM and RSS.
TCP Segmentation and Checksum Offload	[All HCAs] Added TCP segmentation and checksum offload support for MPLS-tagged traffic.
Bug Fixes	See Bug Fixes .

Category	Description
Rev 4.7-3.2.9.0	
Uplink Represor Modes	Added support for the following Uplink Represor modes: <ol style="list-style-type: none"> <code>new_netdev</code> : default mode - when found in this mode, the uplink represor is created as a new netdevice <code>nic_netdev</code> : when found in this mode, the NIC netdevice acts as an uplink represor device Example: <pre>echo nic_netdev > /sys/class/net/ens1f0/compat/devlink/uplink_rep_mode</pre> Notes: <ul style="list-style-type: none"> The mode can only be changed when found in Legacy mode The mode is not saved when reloading <code>mlx5_core</code>
mlx5_core	Added new <code>mlx5_core</code> module parameter " <code>num_of_groups</code> ", which controls the number of large groups in the FDB flow table. Note: In MLNX_OFED v4.6-3.1.9.0.14, the default value of <code>num_of_groups</code> was 15, while in the current MLNX_OFED v4.7-3, the default value is 4. In order to achieve the same OOB experience, make sure to set the <code>num_of_groups</code> module parameter to 15 prior to driver load. For further information, please refer to Performance Tuning Based on Traffic Patterns section in MLNX_OFED User Manual.

Category	Description
Rev 4.7-3.2.9.0	
VFs Groups Minimum Bandwidth Rate	Added support for setting a minimum bandwidth rate on a group of VFs (BW guarantee) to ensure this group is able to transmit at least the amount of bandwidth specified on the wire.
Direct Verbs Support for Batch Counters on Root Table	Added support for mlx5dv_dr API to set batch counters for root tables.
Modify Header	Added support for mlx5dv_dr_actions to support up to 32 modify actions.
mlx5dv_dr Memory Consumption	Reduced the mlx5dv_dr API memory consumption by improving the memory allocator.
mlx5dv_dr Memory Allocation	Reduced memory allocation time when using the mlx5dv_dr API. This is particularly significant for the first inserted rules on which memory is allocated.
Mediated Devices	Added support for mediated devices that allows the creation of accelerated devices without SR-IOV on the Bluefield® system. For further information on mediated devices and how to configure them, please refer to Mediated Devices section in MLNX_EN User Manual.

Category	Description
Rev 4.7-1.0.0.1	
Counters Monitoring	[ConnectX-4 and above] Added support for monitoring selected counters and generating a notification event (Monitor_Counter_Change event) upon changes made to these counters. The counters to be monitored are selected using the SET_MONITOR_COUNTER command.
EEPROM Device Thresholds via Ehttool	[ConnectX-4 and above] Added support to read additional EEPROM information from high pages of modules such as SFF-8436 and SFF-8636. Such information can be: 1. Application Select table 2. User writable EEPROM 3. Thresholds and alarms - Ehttool dump works on active cables only (e.g. optic), but thresholds and alarms can be read with “offset” and “length” parameters in any cable by running: <code>ehttool -m <DEVNAME> offset X length Y</code>
RDMA_RX RoCE Steering Support	[ConnectX-4 and above] Added the ability to create rules to steer RDMA traffic, with two destinations supported: DevX object and QP. Multiple priorities are also supported.
ASAP ²	[ConnectX-5 and above] Incorporated the documentation of <i>Accelerated Switching And Packet Processing (ASAP²): Hardware Offloading for vSwitches</i> into MLNX_OFED Release Notes and User Manual.
MLNX_OFED Installation via Repository	[All HCAs] The repository providing legacy verbs has been moved from RPMS or DEBS folders to RPMS/MLNX_LIBS and DEBS/MLNX_LIBS. In addition, a new repository providing RDMA-Core based userspace has been added to RPMS/UPSTREAM_LIBS and DEBS/UPSTREAM_LIBS.

Category	Description
Rev 4.6-1.0.1.1	
Devlink Configuration Parameters Tool	[ConnectX-3/ConnectX-3 Pro] Added support for a set of configuration parameters that can be changed by the user through the Devlink user interface.
ODP Pre-fetch	[ConnectX-4 and above] Added support for pre-fetching a range of an on-demand paging (ODP) memory region (MR), this way reducing latency by making pages present with RO/RW permissions before the actual IO is conducted.
DevX Privilege Enforcement	[ConnectX-4 and above] Enforced DevX privilege by firmware. This enables future device functionality without the need to make driver changes unless a new privilege type is introduced.
DevX Interoperability APIs	[ConnectX-4 and above] Added support for modifying and/or querying for a verb object (including CQ, QP, SRQ, WQ, and IND_TBL APIs) via the DevX interface. This enables interoperability between verbs and DevX.
DevX Asynchronous Query Commands	[ConnectX-4 and above][ConnectX-4 and above] Added support for running QUERY commands over the DevX interface in an asynchronous mode. This enables applications to issue many commands in parallel while firmware processes the commands.
DevX User-space PRM Handles Exposure	[ConnectX-4 and above] Exposed all PRM handles to user-space so DevX user application can mix verbs objects with DevX objects. For example: Take the cq from the created <code>ibv_cq</code> and use it on a <code>devx)create(QP)</code> .
Indirect Mkey ODP	[ConnectX-4 and above] Added the ability to create indirect Mkeys with ODP support over DevX interface.
XDP Redirect	[ConnectX-4 and above] Added support for XDP_REDIRECT feature for both ingress and egress sides. Using this feature, incoming packets on one interface can be redirected very quickly into the transmission queue of another capable interface. Typically used for load balancing.
RoCE Disablement	[ConnectX-4 and above] Added the option to disable RoCE traffic handling. This enables forwarding of traffic over UDP port 4791 that is handled as RoCE traffic when RoCE is enabled. When RoCE is disabled, there is no GID table, only Raw Ethernet QP type is supported and RoCE traffic is handled as regular Ethernet traffic.
Forward Error Correction (FEC) Encoding	[ConnectX-4 and above] Added the ability to query and modify Forward Error Correction (FEC) encoding, as well as disabling it via Ethtool.
RAW Per-Lane Counters Exposure	[ConnectX-4 and above] Exposed RAW error counters per cable-module lane via ethtool stats. The counters show the number of errors before FEC correction (if enabled). For further information, please see <code>phy_raw_errors_lane[i]</code> under Physical Port Counters section in Understanding mlx5 ethtool Counters Community post.
VF LAG	[ConnectX-4 Lx and above] Added support for High Availability and load balancing for Virtual Functions of different physical ports in SwitchDev SR-IOV mode.
ASAP ² Offloading VXLAN Decapsulation with HW LRO	[ConnectX-5 and above] Added support for performing hardware Large Receive Offload (HW LRO) on VFs with HW-decapsulated VXLAN. For further information on the VXLAN decapsulation feature, please refer to ASAP ² User Manual under nvidia.com/en-us/networking/.com → Products → Software → ASAP ² .
PCI Atomic Operations	[ConnectX-5 and above] Added the ability to run atomic operations on local memory without involving verbs API or compromising the operation's atomicity.
Virtual Ethernet Port Aggregator (VEPA)	[ConnectX-5] Added support for activating/deactivating Virtual Ethernet Port Aggregator (VEPA) mode on a single virtual function (VF). To turn on VEPA on the second VF, run: <code>echo ON > /sys/class/net/enp59s0/device/sriov/1/vepa</code>

Category	Description
Rev 4.6-1.0.1.1	
VFs Rate Limit	[ConnectX-5] Added support for setting a rate limit on groups of Virtual Functions rather than on an individual Virtual Function.
ConnectX-6 Support	[ConnectX-6] [Beta] Added support for ConnectX-6 (VPI only) adapter cards. NOTE: In HDR installations that are built with remotely managed Quantum-based switches, the switch's firmware must be upgraded to version 27.2000.1142 prior to upgrading the HCA's (ConnectX-6) firmware to version 20.25.1500. When using ConnectX-6 HCAs with firmware v20.25.1500 and connecting them to Quantum-based switches, make sure the Quantum firmware version is 27.2000.1142 in order to avoid any critical link issues.
Ethtool 200Gbps	[ConnectX-6] ConnectX-6 hardware introduces support for 200Gbps and 50Gbps-per-lane link mode. MLNX_OFED supports full backward compatibility with previous configurations. Note that in order to advertise newly added link-modes, the full bitmap related to the link modes must be advertised from ethtool man page. For the full bitmap list per link mode, please refer to MLNX_OFED User Manual. NOTE: This feature is firmware-dependent. Currently, ConnectX-6 Ethernet firmware supports up to 100Gbps only. Thus, this capability may not function properly using the current driver and firmware versions.
PCIe Power State	[ConnectX-6] Added support for the following PCIe power state indications to be printed to dmesg: <ol style="list-style-type: none"> Info message #1: PCIe slot power capability was not advertised. Warning message: Detected insufficient power on the PCIe slot (xxxW). Info message #2: PCIe slot advertised sufficient power (xxxW). When indication #1 or #2 appear in dmesg, user should make sure to use a PCIe slot that is capable of supplying the required power.
Message Signaled Interrupts-X (MSI-X) Vectors	[mlx5 Driver] Added support for using a single MSI-X vector for all control event queues instead of one MSI-X vector per queue in a virtual function driver. This frees extra MSI-X vectors to be used for completion event queue, allowing for additional traffic channels in the network device.
Send APIs	[mlx5 Driver] Introduced a new set of QP Send operations (APIs) which allows extensibility for new Send opcodes.
BlueField Support	[BlueField] BlueField is now fully supported as part of the NVIDIA OFED mainstream version sharing the same code baseline with all the adapters product line.
Representor Name Change	[BlueField] In SwitchDev mode: <ul style="list-style-type: none"> Uplink representors are now called p0/p1 Host PF representors are now called pf0hpf/pf1hpf VF representors are now called pf0vfN/pf1vfN
ECPF Net Devices	[BlueField] In SwitchDev mode, net devices enp3s0f0 and enp3s0f1 are no longer created.
Setting Host MAC and Tx Rate Limit from ECPF	[BlueField] Expanded to support VFs as well as the host PFs.
RDMA-CM Application Managed QP	[All HCAs] Added support for the RDMA application to manage its own QPs and use RDMA-CM only for exchanging Address information.
RDMA-CM QP Timeout Control	[All HCAs] Added a new option to <code>rdma_set_option</code> that allows applications to override the RDMA-CM's QP ACK timeout value.

Category	Description
Rev 4.6-1.0.1.1	
MLNX_OFED Verbs API	[All HCAs] As of MLNX_OFED v5.0 release (Q1 2020) onwards, MLNX_OFED Verbs API will be migrated from the legacy version of the user space verbs libraries (libibverbs, libmlx5 ..) to the upstream version rdma-core. More details are available in MLNX_OFED user manual under Installing Upstream rdma-core Libraries .

Category	Description
4.5-1.0.1.0	
VFs per PF	[ConnectX-5] Increased the amount of maximum virtual functions (VF) that can be allocated to a physical function (PF) to 127 VF.
SW-Defined UDP Source Port for RoCE v2	[ConnectX-4/ConnectX-4 Lx/ConnectX-5] UDP source port for RoCE v2 packets is now calculated by the driver rather than the firmware, achieving better distribution and less congestion. This mechanism works for RDMA- CM QPs only, and ensures that RDMA connection messages and data messages have the same UDP source port value.
Local Loopback Disable	[mlx5 Driver] Added the ability to manually disable Local Loopback regardless of the number of open user-space transport domains.
Adapter Cards	[ConnectX-6] Added support for ConnectX-6 Ready. For further information, please contact NVIDIA Support .
Bug Fixes	See Bug Fixes .

4.4-2.0.7.0	
Operating Systems	[All HCAs] Added support for additional OSs. See " General Support " section.

4.4-1.0.1.0	
Adaptive Interrupt Moderation	[ConnectX-4/ConnectX-4 Lx/ConnectX-5] Added support for adaptive Tx, which optimizes the moderation values of the Tx CQs on runtime for maximum throughput with minimum CPU overhead. This mode is enabled by default. [ConnectX-4/ConnectX-4 Lx/ConnectX-5] Updated Adaptive Rx to ignore ACK packets so that queues that only handle ACK packets remain with the default moderation.
Docker Containers [Beta]	[ConnectX-4/ConnectX-4 Lx/ConnectX-5] Added support for Docker containers to run over Virtual RoCE and InfiniBand devices using SR-IOV mode.
Firmware Tracer	[ConnectX-4/ConnectX-4 Lx/ConnectX-5] Added a new mechanism for the device's FW/ HW to log important events into the event tracing system (/sys/kernel/debug/tracing) without requiring any NVIDIA-specific tool. Note: This feature is enabled by default.
CR-Dump	[ConnectX-4/ConnectX-4 Lx/ConnectX-5] Accelerated the original cr-dump by optimizing the reading process of the device's CR-Space snapshot.
VST Q-in-Q	[ConnectX-4/ConnectX-4 Lx] Added support for C-tag (0x8100) VLAN insertion to tagged packets in VST mode.

OVS Offload using ASAP2	[ConnectX-4 Lx/ConnectX-5] Added support for NVIDIA Accelerated Switching And Packet Processing (ASAP2) technology, which allows OVS offloading by handling OVS data-plane, while maintaining OVS control-plane unmodified. OVS Offload using ASAP2 technology provides significantly higher OVS performance without the associated CPU load. For further information, refer to ASAP2 Release Notes under nvidia.com/en-us/networking/.com → Products → Software → ASAP ² .
4.3-1.0.1.0	
Adaptive Interrupt Moderation	[ConnectX-4/ConnectX-4 Lx/ConnectX-5] Added support for adaptive Tx, which optimizes the moderation values of the Tx CQs on runtime for maximum throughput with minimum CPU overhead. This mode is enabled by default.
	[ConnectX-4/ConnectX-4 Lx/ConnectX-5] Updated Adaptive Rx to ignore ACK packets so that queues that only handle ACK packets remain with the default moderation.
Docker Containers [Beta]	[ConnectX-4/ConnectX-4 Lx/ConnectX-5] Added support for Docker containers to run over Virtual RoCE and InfiniBand devices using SR-IOV mode.
Firmware Tracer	[ConnectX-4/ConnectX-4 Lx/ConnectX-5] Added a new mechanism for the device's FW/HW to log important events into the event tracing system (/sys/kernel/debug/tracing) without requiring any NVIDIA-specific tool. Note: This feature is enabled by default.
CR-Dump	[ConnectX-4/ConnectX-4 Lx/ConnectX-5] Accelerated the original cr-dump by optimizing the reading process of the device's CR-Space snapshot.
VST Q-in-Q	[ConnectX-4/ConnectX-4 Lx] Added support for C-tag (0x8100) VLAN insertion to tagged packets in VST mode.
OVS Offload using ASAP2	[ConnectX-4 Lx/ConnectX-5] Added support for NVIDIA Accelerated Switching And Packet Processing (ASAP2) technology, which allows OVS offloading by handling OVS data-plane, while maintaining OVS control-plane unmodified. OVS Offload using ASAP2 technology provides significantly higher OVS performance without the associated CPU load. For further information, refer to ASAP2 Release Notes under nvidia.com/en-us/networking/.com → Products → Software → ASAP ² .
4.3-1.0.1.0	
Erasur Coding Offload verbs	[ConnectX-5] Added support for erasure coding offload software verbs (encode/decode/update API) supporting a number of redundancy blocks (m) greater than 4.
Virtual MAC	[ConnectX-4/ConnectX-4 Lx/ConnectX-5] Removed support for Virtual MAC feature.
RoCE LAG	[ConnectX-4/ConnectX-4 Lx/ConnectX-5] Added out of box RoCE LAG support for RHEL 7.2 and RHEL 6.9.
Dropped Counters	[ConnectX-4/ConnectX-4 Lx/ConnectX-5] Added a new counter <i>rx_steer_missed_packets</i> which provides the number of packets that were received by the NIC, yet were discarded/dropped since they did not match any flow in the NIC steering flow table.
	[ConnectX-4/ConnectX-4 Lx/ConnectX-5] Added the ability for SR-IOV counter <i>rx_dropped</i> to count the number of packets that were dropped while vport was down.
Reset Flow	[mlx5 Driver] Added support for triggering software reset for firmware/driver recovery. When fatal errors occur, firmware can be reset and driver reloaded.
Striding RQ with HW Time-Stamping	[ConnectX-4 Lx/ConnectX-5] Added the option to retrieve the HW timestamp when polling for completions from a completion queue that is attached to a multi-packet RQ (Striding RQ).
4.2-1.0.1.0	

Physical Address Memory Allocation	[mlx5 Driver] Added support to register a specific physical address range.
Innova IPsec Adapter Cards	[Innova IPsec EN] Added support for NVIDIA Innova IPsec EN adapter card, that provides security acceleration for IPsec-enabled networks.
Precision Time Protocol (PTP)	[ConnectX-4/ConnectX-4 Lx/ConnectX-5] Added support for PTP feature over PKEY interfaces. This feature allows for accurate synchronization between the distributed entities over the network. The synchronization is based on symmetric Round Trip Time (RTT) between the master and slave devices, and is enabled by default.
Virtual MAC	[ConnectX-4/ConnectX-4 Lx/ConnectX-5] Added support for Virtual MAC feature, which allows users to add up to 4 virtual MACs (VMACs) per VF. All traffic that is destined to the VMAC will be forwarded to the relevant VF instead of PF. All traffic going out from the VF with source MAC equal to VMAC will go to the wire also when Spoof Check is enabled. For further information, please refer to “Virtual MAC” section in MLNX_EN User Manual.
Receive Buffer	[ConnectX-4/ConnectX-4 Lx/ConnectX-5] Added the option to change receive buffer size and cable length. Changing cable length will adjust the receive buffer's xon and xoff thresholds. For further information, please refer to “Receive Buffer” section in MLNX_EN User Manual.
GRE Tunnel Offloads	[ConnectX-4/ConnectX-4 Lx/ConnectX-5] Added support for the following GRE tunnel offloads: <ul style="list-style-type: none"> • TSO over GRE tunnels • Checksum offloads over GRE tunnels • RSS spread for GRE packets
NVMeoF	[ConnectX-4/ConnectX-4 Lx/ConnectX-5] Added support for the host side (RDMA initiator) in RedHat 7.2 and above.
Droplless Receive Queue (RQ)	[ConnectX-4/ConnectX-4 Lx/ConnectX-5] Added support for the driver to notify the FW when SW receive queues are overloaded.
PFC Storm Prevention	[ConnectX-4/ConnectX-4 Lx/ConnectX-5] Added support for configuring PFC stall prevention in cases where the device unexpectedly becomes unresponsive for a long period of time. PFC stall prevention disables flow control mechanisms when the device is stalled for a period longer than the default pre-configured timeout. Users now have the ability to change the default timeout by moving to auto mode. For further information, please refer to “PFC Stall Prevention” section in MLNX_EN User Manual.
Q-in-Q	[ConnectX-5] Added support for Q-in-Q VST feature in ConnectX-5 adapter cards family.
Virtual Guest Tagging (VGT+)	[ConnectX-5] Added support for VGT+ in ConnectX-4/ConnectX-5 HCAs. This feature is s an advanced mode of Virtual Guest Tagging (VGT), in which a VF is allowed to tag its own packets as in VGT, but is still subject to an administrative VLAN trunk policy. The policy determines which VLAN IDs are allowed to be transmitted or received. The policy does not determine the user priority, which is left unchanged. For further information, please refer to “Virtual Guest Tagging (VGT+)” section in MLNX_EN User Manual.
Tag Matching Offload	[ConnectX-5] Added support for hardware Tag Matching offload with Dynamically Connected Transport (DCT).
CR-DUMP	[All HCAs] Added support for the driver to take an automatic snapshot of the device's CR-Space in cases of critical failures. For further information, please refer to “CRDUMP” section in MLNX_EN User Manual.
4.1-1.0.2.0	

RoCE Diagnostics and ECN Counters	[mlx5 Driver] Added support for additional RoCE diagnostics and ECN congestion counters under /sys/class/infiniband/mlx5_0/ports/1/hw_counters/ directory. For further information, refer to the Understanding mlx5 Linux Counters and Status Parameters Community post.
rx-fcs Offload (ethtool)	[mlx5 Driver] Added support for rx-fcs ethtool offload configuration. Normally, the FCS of the packet will be truncated by the ASIC hardware before sending it to the application socket buffer (skb). Ethtool allows to set the rx-fcs not to be truncated, but to pass it to the application for analysis. For more information and usage, refer to Understanding ethtool rx-fcs for mlx5 Drivers Community post.
DSCP Trust Mode	[mlx5 Driver] Added the option to enable PFC based on the DSCP value. Using this solution, VLAN headers will no longer be mandatory for use. For further information, refer to the HowTo Configure Trust Mode on NVIDIA Adapters Community post.
RoCE ECN Parameters	[mlx5 Driver] ECN parameters have been moved to the following directory: /sys/kernel/debug/mlx5/<PCI BUS>/cc_params/ For more information, refer to the HowTo Configure DCQCN (RoCE CC) for ConnectX-4 (Linux) Community post.
Flow Steering Dump Tool	[mlx5 Driver] Added support for mlx_fs_dump, which is a python tool that prints the steering rules in a readable manner.
Secure Firmware Updates	[mlx5 Driver] Firmware binaries embedded in MLNX_EN package now support Secure Firmware Updates. This feature provides devices with the ability to verify digital signatures of new firmware binaries, in order to ensure that only officially approved versions are installed on the devices. For further information on this feature, refer to NVIDIA Firmware Tools (MFT) User Manual.
PeerDirect	[mlx5 Driver] Added the ability to open a device and create a context while giving PCI peer attributes such as name and ID. For further details, refer to the PeerDirect Programming Community post.
Probed VFs	[mlx5 Driver] Added the ability to disable probed VFs on the hypervisor. For further information, see HowTo Configure and Probe VFs on mlx5 Drivers Community post.
Local Loopback	[mlx5 Driver] Improved performance by rendering Local loopback (unicast and multicast) disabled by mlx5 driver by default while local loopback is not in use. The mlx5 driver keeps track of the number of transport domains that are opened by user-space applications. If there is more than one user-space transport domain open, local loopback will automatically be enabled.
1PPS Time Synchronization (at alpha level)	[mlx5 Driver] Added support for One Pulse Per Second (1PPS), which is a time synchronization feature that allows the adapter to send or receive 1 pulse per second on a dedicated pin on the adapter card. For further information on this feature, refer to the HowTo Test 1PPS on NVIDIA Adapters Community post.
Fast Driver Unload	[mlx5 Driver] Added support for fast driver teardown in shutdown and kexec flows.
NVMeoF Target Offload	[ConnectX-5/ConnectX-5 Ex] Added support for NVMe over fabrics (NVMeoF) offload, an implementation of the new NVMeoF standard target (server) side in hardware. For further information on NVMeoF Target Offload, refer to HowTo Configure NVMeoF Target Offload .
RDMA CM	[All HCAs] Changed the default RoCE mode on which RDMA CM runs to RoCEv2 instead of RoCEv1. RDMA_CM session requires both the client and server sides to support the same RoCE mode. Otherwise, the client will fail to connect to the server. For further information, refer to RDMA CM and RoCE Version Defaults Community post.
Lustre	[All HCAs] Added support for Lustre file system open-source project.

4.0-2.0.0.1	
PCIe Error Counting	[ConnectX-4/ConnectX-4 Lx] Added the ability to expose physical layer statistical counters to ethtool.
Standard ethtool	[ConnectX-4/ConnectX-4 Lx] Added support for flow steering and rx-all mode.
SR-IOV Bandwidth Share for Ethernet/RoCE (beta)	[ConnectX-4/ConnectX-4 Lx] Added the ability to guarantee the minimum rate of a certain VF in SR-IOV mode.
Adapter Cards	Added support for ConnectX-5 and ConnectX-5 Ex HCAs.
NFS over RDMA (NFSv4.1)	Removed support for NFSv4.1 drivers. These drivers are no longer provided along with the MLNX_EN package.

3.1.1.2 Customer Affecting Changes

Customer Affecting Change	Description
Customer Affecting Changes 5.6-1.0.3.5	
Interface Renaming, PF/VF, Udev	<p>The OFED driver no longer performs Ethernet NetDev interface renaming for PFs and VFs.</p> <p>The udev rules file which implemented renaming (82-net-setup-link.rules) and its supporting script vf-net-link-name.sh are no longer installed by default. Renaming is thus performed by underlying mechanisms -- in udev, in the kernel, and in the BIOS.</p> <p>Users who wish to continue using the OFED driver renaming mechanism must add option <code>--copy-ifnames-udev</code> to the OFED install command.</p> <p>To install these files at a later time, copy them from one of the following directories:</p> <ul style="list-style-type: none"> • /usr/share/doc/mlnx-ofa_kernel (RHEL8 and newer) • /usr/share/doc/mlnx-ofa_kernel-[1-9]* (RHEL 7.X) • /usr/share/doc/packages/mlnx-ofa_kernel (SLES) • /usr/share/doc/mlnx-qed-kernel-utils/examples (Debian-based releases) <div style="border: 1px solid orange; padding: 10px; margin-top: 10px;"> <ul style="list-style-type: none"> • File 82-net-setup-link.rules should be copied to directory /etc/udev/rules.d • File vf-net-link-name.sh should be copied to directory /etc/infiniband (make sure that it has both read and execute permission) • After copying over the files, the driver should be restarted for the copied files to take effect • Customers who wish prevent renaming of NetDev names should add "net.ifnames=0 biosdevname=0" to the kernel boot command line, and then reboot the host </div>
Community Operating Systems	<p>Starting OFED 5.6, NVIDIA is introducing a new support model for OFED used on open source community operating systems. The goal of this new support model is to enable customers to use community-maintained variants of the Linux operating system, without being limited to major distributions that NVIDIA provides primary support for. For more information, see "Installation on Community Operating Systems" section in the user manual. For a list of supported Community OSs, please see "Supported Community Operating Systems" section in the release notes.</p>

Customer Affecting Change	Description
Customer Affecting Changes 5.6-1.0.3.5	
OVS-DPDK–Partial Offload	Starting OFED 5.6, OVS-DPDK does not support partial offload.

Customer Affecting Change	Description
5.5-1.0.3.2	
Disabling RoCE While Using sysfs	When using sysfs to enable/disable roce in kernel 5.5 and up, the "devlink reload" command (using iproute2 with devlink tool) will need to be used to activate the RoCE status change. Disable RoCE example: 1. echo 0 > /sys/bus/pci/devices/0000:08:00.0/roce_enable 2. devlink dev reload pci/0000:08:00.0
mlnx-ofa_kernel Installation	The source code for mlnx-ofa_kernel is no longer installed by default on RPM-based distributions (e.g., RHEL and SLES). Notes: <ul style="list-style-type: none"> • mlnx-ofa_kernel is included in the <<package mlnx-ofa_kernel-devel-source>> in the MLNX_OFED distributions under RPMS/ and may be manually installed from there. • There is no change for deb-based distributions (Debian and Ubuntu). The full source is included, as before, in the package mlnx-ofed-kernel-dkms.
Software Encapsulation Compatibility	There is an encapsL2 compatibility issue with accelerated reformat action creation using mlx5dv_dr API. Using OFED 5.4 with firmware xx.32.1xxx and above or using OFED 5.5 with firmware lower than xx.32.1xxx will not allow accelerated reformat action. (Using OFED 5.4 and 5.5 with bundle firmware works properly.)
xpmem in RHEL8	Added xpmem packages in RHEL8 builds.
Python3	Starting OVS DPDK 2.15, the Python minimum required version is 3 and OVS-DPDK will not be compiled using Python 2.

3.1.2 Bug Fixes History

This table lists the bugs fixed in the last three major GA releases. For a list of old bug fixes, please refer to the release notes of the desired version.

Fixed in Release: 5.8-3.0.7.0

Internal Reference Number	Description
3344682	Description: If there are multiple encapsulations and not all neighbors are valid, the kernel will go into panic mode.
	Keywords: ASAP ² , Kernel Panic
	Fixed in Release: 5.8-2.0.3.0

Internal Reference Number	Description
3350185	Description: IRQ naming was incorrect for mlx5 interfaces. From now on, the IRQ naming on an inactive channel will be indexed from 0-(n-1).
	Keywords: Core, IRQ Naming
	Fixed in Release: 5.8-2.0.3.0
3333920	Description: Changing traffic class via the sysfs while modifying QPs in parallel, causes a deadlock.
	Keywords: RDMA, TC, Sysfs, QP
	Fixed in Release: 5.8-2.0.3.0

Internal Reference Number	Description
3253500	Description: The redundant freeing of a list item could lead to memory corruption, potentially causing the application to crash or incorrect traffic handling.
	Keywords: Steering, Memory Corruption, List, Pattern/Argument
	Fixed in Release: 5.8-1.1.2.1
3214161	Description: The knem-dkms package explicitly requires GCC to build the knem driver (at install times). Under some circumstances, on Debian systems, the apt install method may result in a system that has only gcc-<version> (e.g., gcc-10) installed.
	Keywords: Installation, Debian, GCC
	Fixed in Release: 5.8-1.1.2.1
3230613	Description: Installing MLNX_OFED_LINUX on an Ubuntu system with CUDA (version < 11.6) may result in an automatic installation of the ucx-cuda package that will fail with an error message in the log file ucx-cuda.deinstall.log about missing dependencies.
	Keywords: Installation, Ubuntu, CUDA
	Fixed in Release: 5.8-1.1.2.1
3235521	Description: The host driver probe did not check whether there are existing SFs which are present in the device, causing the host driver to not recreate those SFs.
	Keywords: Core, Scalable Functions
	Fixed in Release: 5.8-1.1.2.1
3228357	Description: If there are multiple encapsulations and not all neighbors are valid, the kernel will go into panic mode.
	Keywords: ASAP ² , Encapsulation
	Discovered in Release: 5.5-1.0.3.2, 5.7-1.0.2.0
	Fixed in Release: 5.8-1.1.2.1
3232445	Description: When using BlueField with old kernels, multiple OVS meter do not work.
	Keywords: ASAP ² , BlueField, Meter, OVS, Offload
	Fixed in Release: 5.8-1.1.2.1

Internal Reference Number	Description
3234066	<p>Description: When configuring IPsec full offload, after sending traffic for approximately 30 minutes, the traffic stops at some point and the connection gets lost.</p> <p>Keywords: Steering, SMFS, Matcher Disconnect</p> <p>Fixed in Release: 5.8-1.0.1.1</p>
3179535	<p>Description: SMFS will try to merge flow rules with the same matching criteria (as they share the same matcher) into one multi-destination rule. If merging fails, the matcher is disconnected by mistake.</p> <p>Keywords: Steering, SMFS, Matcher Disconnect</p> <p>Fixed in Release: 5.8-1.0.1.1</p>
3214198	<p>Description: ibv_reg_mr for huge pages was optimized in kernel >= 5.12</p> <p>Keywords: RDMA, ibv_reg_mr</p> <p>Discovered in Release: 5.7-1.0.2.0</p> <p>Fixed in Release: 5.8-1.0.1.1</p>
2984134	<p>Description: Moving to SwitchDev mode while deleting namespace over Linux-6.0 can sometimes cause a deadlock.</p> <p>Keywords: RDMA, SwitchDev</p> <p>Discovered in Release: 5.5-1.0.3.2</p> <p>Fixed in Release: 5.8-1.0.1.1</p>
3106228	<p>Description: A net device validation issue prevented running IPv6 traffic using an RDMA communication manager between two interfaces on same host with same subnet.</p> <p>Keywords: RDMA, IPv6, Communication Manager</p> <p>Discovered in Release: 5.6-1.0.3.3</p> <p>Fixed in Release: 5.8-1.0.1.1</p>
3151843	<p>Description: In mlx5dv_mkey_check manpage, there is an inaccurate description of signature error handling flow.</p> <p>Keywords: RDMA, manpage</p> <p>Discovered in Release: 5.7-1.0.2.0</p> <p>Fixed in Release: 5.8-1.0.1.1</p>
3229002	<p>Description: Creating and deleting MRs, caused a kernel slab cache leak issue.</p> <p>Keywords: RDMA, Cache</p> <p>Discovered in Release: 5.7-1.0.2.0</p> <p>Fixed in Release: 5.8-1.0.1.1</p>
3236217	<p>Description: The <code>rdma res show cm_id</code> command does not list all cm_ids when some of them are in LISTEN state.</p> <p>Keywords: RDMA, cm_ids</p> <p>Discovered in Release: 5.0-1.0.0.0</p> <p>Fixed in Release: 5.8-1.0.1.1</p>
3146128	<p>Description: In older kernel version, PTP was not supported over VLAN interfaces.</p>

Internal Reference Number	Description
	<p>Keywords: NetDev, PTP, VLAN</p> <p>Discovered in Release: 5.7-1.0.2.0</p> <p>Fixed in Release: 5.8-1.0.1.1</p>
2969772	<p>Description: HW-GRO feature was blocked due to firmware limitations.</p> <p>Keywords: NetDev, HW-GRO</p> <p>Discovered in Release: 5.6-1.0.3.3</p> <p>Fixed in Release: 5.8-1.0.1.1</p>
3096393	<p>Description: STP packets failed to be transmitted.</p> <p>Keywords: NetDev, STP</p> <p>Discovered in Release: 5.5-1.0.3.2</p> <p>Fixed in Release: 5.8-1.0.1.1</p>
3236984	<p>Description: When using sysfs to read the hash function used to distribute the traffic between the TIRs (Transport Interface Receive), on occasion, the server crashed.</p> <p>Keywords: NetDev, sysfs</p> <p>Discovered in Release: 5.7-1.0.2.0</p> <p>Fixed in Release: 5.8-1.0.1.1</p>
3126000	<p>Description: Upgrading from version 5.6-2 to 5.7 failed.</p> <p>Keywords: Installation</p> <p>Discovered in Release: 5.6-2.0.9.0</p> <p>Fixed in Release: 5.8-1.0.1.1</p>
3230524	<p>Description: Building with KMP enabled fails due to missing packages. OFED packages will now be built with KMP disabled.</p> <p>Keywords: Installation, KMP</p> <p>Fixed in Release: 5.8-1.0.1.1</p>
3158725	<p>Description: The script install.pl, used for (re)building kernel modules, used the name "kernel-source" as the package of the kernel-source on SLES systems.</p> <p>Keywords: Installation, SLES</p> <p>Discovered in Release: 5.6-1.0.3.3</p> <p>Fixed in Release: 5.8-1.0.1.1</p>
3142212	<p>Description: Starting firmware version xx.34.0350, a new NVCONFIG has been added to the ARM side only: MANAGEMENT_PF_MODE. If this config is on, the user will see a PCI Function (PF) which failed to probe:</p> <pre data-bbox="443 1727 1390 1921"> [6.837102] mlx5_core 0000:03:00.2: mlx5_cmd_check:756:(pid 206): ENABLE_HCA(0x104) op_mod(0x0) failed, status bad parameter(0x3), syndrome (0x6ca1f5) [6.864227] mlx5_core 0000:03:00.2: mlx5_peer_pf_init:40:(pid 206): Failed to enable peer PF HCA err(-22) [6.883453] mlx5_core 0000:03:00.2: mlx5_load:1129:(pid 206): Failed to init embedded CPU [8.261268] mlx5_core 0000:03:00.2: init_one:1365:(pid 206): mlx5_load_one failed with error code -22 [8.280056] mlx5_core: probe of 0000:03:00.2 failed with error -22 </pre>

Internal Reference Number	Description
	Keywords: Installation
	Discovered in Release: 5.7-1.0.2.0
	Fixed in Release: 5.8-1.0.1.1
3174928	Description: Using a 1-CPU system casues possible command flush deadlock.
	Keywords: Core
	Discovered in Release: 5.6-1.0.3.3
	Fixed in Release: 5.8-1.0.1.1
3228721/3228357	Description: An incorrect termination table was used with the uplink-to-uplink forward rule.
	Keywords: ASAP ² , eSwitch
	Discovered in Release: 5.7-1.0.2.0
	Fixed in Release: 5.8-1.0.1.1
3220120	Description: In old kernels, when a VXLAN tunnel is set up on one OVS bridge and PF is up on another OVS bridge, traffic does not offload as expected.
	Keywords: ASAP ² , VXLAN
	Discovered in Release: 5.4-3.0.3.0
	Fixed in Release: 5.8-1.0.1.1

Internal Reference Number	Description
3032335	Description: Creating multiple steering rules that modify a packet and match on the same packet headers can cause an error to be displayed in dmesg when deleting the steering rules.
	Keywords: Steering Rules
	Fixed in Release: 5.7-1.0.2.0
3011368	Description: Some IB spec QP state behaviour on post_send()/recv() is not being fully enforced. The fix makes the QP complaint to IB spec about when it is allowed to post_send()/recv() and when it should return an error.
	Keywords: RDMA, IB spec QP
	Fixed in Release: 5.7-1.0.2.0
3075125	Description: When changing trust state from PCP to DSCP, the TC number changes by default to 8, in some cases, disrupting traffic prioritization if trust state is changed back to PCP.
	Keywords: NetDev, QoS
	Discovered in Release: 5.4-1.0.3.0
	Fixed in Release: 5.7-1.0.2.0

Internal Reference Number	Description
3054413	<p>Description: In the current release, the following OPNs/PSIDs should be manually upgraded: MCX753106AS-HEA-N NVD0000000023 MCX75310AAS-HEA-N NVD0000000024</p> <p>Keywords: ConnectX-7, Upgrade</p> <p>Discovered in Release: 5.6-1.0.3.3</p> <p>Fixed in Release: 5.7-1.0.2.0</p>
3070653	<p>Description: In versions of MLNX_OFED before 5.7, the xpmem kernel module was not signed. When it was installed on systems (mostly RHEL and other compatible systems) the following error message would appear: "xpmem: loading out-of-tree module taints kernel."</p> <p>Keywords: Installation, xpmem</p> <p>Discovered in Release: 5.6-1.0.3.3</p> <p>Fixed in Release: 5.7-1.0.2.0</p>
3075357	<p>Description: In Debian-based distributions, in /etc/init.d/openibd, the path to enable the firmware tracer is /sys/kernel/debug/tracing/events/mlx5/fw_tracer/enable instead of /sys/kernel/debug/tracing/events/mlx5/mlx5_fw/enable . As a result, firmware tracer will never get enabled even when supported.</p> <p>Keywords: Installation, Kernel Trace Debug</p> <p>Discovered in Release: 5.6-1.0.3.3</p> <p>Fixed in Release: 5.7-1.0.2.0</p>
2688191	<p>Description: The minimum Tx rate limit is not supported with link speed of 1Gb/s.</p> <p>Keywords: Rate Limit, 1Gb/s</p> <p>Discovered in Release: 5.6-1.0.3.3</p> <p>Fixed in Release: 5.7-1.0.2.0</p>
3044255	<p>Description: Destroying mlxdevm group while SF is attached to it is not supported.</p> <p>Keywords: ASAP², mlxdevm, QoS, Group, Scalable Functions</p> <p>Discovered in Release: 5.6-1.0.3.3</p> <p>Fixed in Release: 5.7-1.0.2.0</p>
3047142	<p>Description: Using OVS offload with NIC mode (non switchdev mode) causes traffic to drop.</p> <p>Keywords: ASAP², Offload, NIC Mode, OVS</p> <p>Discovered in Release: 5.6-1.0.3.3</p> <p>Fixed in Release: 5.7-1.0.2.0</p>
3123986	<p>Description: In some cases VF metering configuration failure caused a deadlock.</p> <p>Keywords: ASAP², VF Metering</p> <p>Discovered in Release: 5.5-1.0.3.2</p> <p>Fixed in Release: 5.7-1.0.2.0</p>

Internal Reference Number	Description
3053842	Description: A race condition may cause some connection aging to set to 24 hours instead of 30 seconds.
	Keywords: ASAP ² , Connection Tracking, Aging
	Discovered in Release: 5.5-1.0.3.2
	Fixed in Release: 5.7-1.0.2.0

Internal Reference Number	Description
3079038	Description: When an already-loaded 'non-mellanox' auxiliary device on the auxiliary bus OFED driver exists, load may fail and cause kernel panic.
	Keywords: Driver Load
	Discovered in Release: 5.6-1.0.3.3
	Fixed in Release: 5.6-2.0.9.0
3066233	Description: On SLES15 systems that have both python3 and python2 installed, rebuilding kernel modules fails with an error in the mlnx-tools package, and specifically in the mlnx-tools build log, about missing ib2ibsetup.8.
	Keywords: Installation
	Discovered in Release: 5.6-1.0.3.3
	Fixed in Release: 5.6-2.0.9.0

Internal Reference Number	Description
2697443	Description: Reloading devlink in NetDev profile caused deadlock.
	Keywords: Devlink
	Discovered in Release: 5.0-1.0.0.0
	Fixed in Release: 5.6-1.0.3.5
2771739	Description: Gratuitous ARP during rdma_connect is not handled properly.
	Keywords: Gratuitous ARP
	Fixed in Release: 5.6-1.0.3.5
2820245	Description: Crypto offload of UDP traffic on top of IPv6 was unsupported.
	Keywords: IPsec, Crypto, Offload
	Discovered in Release: 5.5-1.0.3.2
	Fixed in Release: 5.6-1.0.3.5
2869109	Description: IPsec crypto offload for non TCP/UDP encapsulated traffic broke.
	Keywords: IPsec, Crypto, Offload
	Discovered in Release: 5.5-1.0.3.2

Internal Reference Number	Description
	Fixed in Release: 5.6-1.0.3.5
2905896	<p>Description: Leaving a multicast group (<code>rdma_leave_multicast</code>) used the wrong address and left the interface in the multicast group.</p> <p>Keywords: RoCE, Multicast</p> <p>Discovered in Release: 5.5-1.0.3.2</p> <p>Fixed in Release: 5.6-1.0.3.5</p>
2939691	<p>Description: Unsupported parameters were ignored. Now, when using unsupported syntax or unsupported command line parameters, the application will fail with an error message.</p> <p>Keywords: Command Line, Parameters</p> <p>Fixed in Release: 5.6-1.0.3.5</p>
2939037	<p>Description: Ethtool that is part of the original EN package failed to dump correct EEPROM values when using <code>-m</code> flag.</p> <p>Keywords: Ethtool, EEPROM</p> <p>Discovered in Release: 5.5-1.0.3.2</p> <p>Fixed in Release: 5.6-1.0.3.5</p>
2979137	<p>Description: An increment of count variable was missing when looping over output buffer in <code>mlx5e_self_test()</code>. As a result, the garbage value of ethtool <code>-t</code> was resolved.</p> <p>Keywords: ethtool, selftest</p> <p>Discovered in Release: 5.5-1.0.3.2</p> <p>Fixed in Release: 5.6-1.0.3.5</p>
2752622	<p>Description: On SLES 15, the inbox modules in the directory <code>mlxsw</code> (such as <code>mlxsw_spectrum</code>) was not supported. When they were installed when installing <code>MLNX_EN</code>, they no longer worked (as they depend on a different version of the <code>mlx*</code> modules) and could cause an error at time of installation.</p> <p>Keywords: Installation</p> <p>Discovered in Release: 5.4-3.0.3.0</p> <p>Fixed in Release: 5.6-1.0.3.5</p>
2984013	<p>Description: When uninstalling the <code>kmod-xpmem</code> package, <code>xpmem</code> module was not unloaded. From now on, after uninstalling, <code>xpmem</code> module will be removed automatically.</p> <p>Keywords: Installation, xpmem</p> <p>Discovered in Release: 5.5-1.0.3.2</p> <p>Fixed in Release: 5.6-1.0.3.5</p>
2984098	<p>Description: OFED installation modified file <code>"/etc/yum.conf"</code> to exclude some packages from the Yum repositories. As of RHEL 8, <code>/etc/yum.conf</code> is a symlink to <code>/etc/dnf.conf</code> and this edit breaks the symlink. As there is no use in such an edit, OFED no longer edits this file.</p> <p>Keywords: Installation, Yum Repositories, RHEL</p> <p>Discovered in Release: 5.5-1.0.3.2</p>

Internal Reference Number	Description
	Fixed in Release: 5.6-1.0.3.5
2946450	Description: In some cases, the firmware tracer did not work with NEO-Host.
	Keywords: NEO-Host, Firmware Tracer
	Fixed in Release: 5.6-1.0.3.5
2947645	Description: current_link_speed sysfs was missing.
	Keywords: sysfs
	Discovered in Release: 5.5-1.0.3.2
	Fixed in Release: 5.6-1.0.3.5
3025582	Description: If the commands were not entered in the correct order when setting buffer size and allocation using the mlnx_qos command, on some occasions, the xoff_threshold calculation broke pausing functionality.
	Keywords: Driver, xoff, Buffer
	Discovered in Release: 5.5-1.0.3.2
	Fixed in Release: 5.6-1.0.3.5
2936867	Description: Creating a TC rules with more than 30 actions caused kernel panic.
	Keywords: ASAP ² , Call Trace, TC
	Discovered in Release: 5.5-1.0.3.2
	Fixed in Release: 5.6-1.0.3.5
3016685	Description: IP-in-IP packets received in one queue instead of hashing to multi queues.
	Keywords: NetDev, Tunneling, RSS
	Discovered in Release: 5.4-1.0.3.0
	Fixed in Release: 5.6-1.0.3.5
3023304	Description: Fixed compatibility issue of mlnx_qos for python3.9 deprecated tostring/fromstring.
	Keywords: Python3, Compatibility
	Fixed in Release: 5.6-1.0.3.5
2887387	Description: IPsec flow tables design caused the number of IPsec tunnels to be limited to 16K. Changed the flow tables design to support up to 32K IPsec tunnels per protocol (IPv4/IPv6).
	Keywords: IPsec
	Discovered in Release: 5.4-1.0.3.0
	Fixed in Release: 5.6-1.0.3.5
2887394/2887381	Description: When configuring over 1000 IPsec sessions caused performance issues.
	Keywords: IPsec
	Discovered in Release: 5.5-1.0.3.2
	Fixed in Release: 5.6-1.0.3.5

Internal Reference Number	Description
2906002	Description: Hairpin rules failed to send packet back to wire when IPsec full offload is enabled.
	Keywords: IPsec Full Offload, Hairpin
	Discovered in Release: 5.5-1.0.3.2
	Fixed in Release: 5.6-1.0.3.5
2890024	Description: Under certain conditions, incorrect handling of resources caused memory corruption over software steering resources leading to failure of OVS to offloaded the traffic to the hardware.
	Keywords: ASAP ² , Steering, OVS, Memory
	Discovered in Release: 5.5-1.0.3.2
	Fixed in Release: 5.6-1.0.3.5
2874200	Description: Using hairpin tunnel traffic, caused incorrect TC rules to be created. Example: <pre>tunnel(tun_id=0x65,src=10.10.11.3,dst=10.10.11.2,ttl=0/0,tp_dst=4789,flags(+key)),...,in_port(vxlan_sys_4789),..., actions:set(tunnel(tun_id=0x66,src=10.10.12.2,dst=10.10.12.3,tp_dst=4789,flags(key))),vxlan_sys_4789</pre>
	Keywords: ASAP ² , Hairpin, OVS, SwitchDev
	Discovered in Release: 5.5-1.0.3.2
	Fixed in Release: 5.6-1.0.3.5
2891499	Description: Adding a route with next hop object caused a warning in dmesg and could possibly lead to kernel panic.
	Keywords: ASAP ² , Route, SwitchDev, Call Trace, Nexthop
	Discovered in Release: 5.5-1.0.3.2
	Fixed in Release: 5.6-1.0.3.5

Internal Reference Number	Description
2842077	Description: Between scripts there was a possibility for Inconsistency in python3 header line (shebang line) because some distributions may no longer have /usr/bin/python.
	Keywords: Python3
	Discovered in Release: 5.4-1.0.3.0
	Fixed in Release: 5.5-1.0.3.2
2792432	Description: The driver did not set the PCP-based priority for DCT, hence DCT response packets were transmitted without user priority.
	Keywords: User Priority, DCT
	Fixed in Release: 5.5-1.0.3.2
2792480	Description: Running tcpdump on bonding standby port caused to lose the network.

Internal Reference Number	Description
	<p>Keywords: NetDev</p> <p>Discovered in Release: 5.4-1.0.3.0</p> <p>Fixed in Release: 5.5-1.0.3.2</p>
2782406	<p>Description: Running yum update will upgrade kylin-release to a higher version. The version of this package is used for kylin10sp2 detection so the script will detect kylin 10 instead of kylin10sp2 and use its repository by mistake.</p> <p>Workaround: Upgrade, kylin</p> <p>Discovered in Release: 5.4-3.0.3.0</p> <p>Fixed in Release: 5.5-1.0.3.2</p>
2823700	<p>Description: xpmem driver is not supported on PowerPC.</p> <p>Keywords: Installation, xpmem, PowerPC</p> <p>Fixed in Release: 5.5-1.0.3.2</p>
2802508	<p>Description: Suspend flow freed the VLAN data so the data was not restored during the resume flow.</p> <p>Keywords: VLAN, Suspend Flow, Resume Flow</p> <p>Fixed in Release: 5.5-1.0.3.2</p>
2796010	<p>Description: Connection tracking rules with fragmentation had 0 stats.</p> <p>Keywords: BlueField, Connection Tracking, Fragments, ASAP²</p> <p>Discovered in Release: 5.4-2.4.1.3</p> <p>Fixed in Release: 5.5-1.0.3.2</p>
2803403	<p>Description: Traffic failed to pass when OVS bridge is configured with bond interface and IP is configured over the OVS internal (bridge) port.</p> <p>Keywords: Bond, VF LAG, OVS, Internal Port, ASAP²</p> <p>Discovered in Release: 5.2-1.0.4.0</p> <p>Fixed in Release: 5.5-1.0.3.2</p>
2438392	<p>Description: VXLAN with IPsec crypto offload does not work.</p> <p>Keywords: VXLAN; IPsec crypto</p> <p>Discovered in Release: 5.3-1.0.0.1</p> <p>Fixed in Release: 5.5-1.0.3.2</p>
2677225	<p>Description: Conducting a driver restart while in VF LAG mode may cause unwanted behaviour such as kernel crashes.</p> <p>Keywords: ASAP², Bonding, Driver Restart, VF LAG</p> <p>Discovered in Release: 5.4-1.0.3.0</p> <p>Fixed in Release: 5.5-1.0.3.2</p>

Internal Reference Number	Description
2852904	Description: In version 5.4, there was some offload breakage when using OVS.
	Keywords: TSO, UDP Tunnels
	Discovered in Release: 5.4-1.0.3.0
	Fixed in Release: 5.4-3.1.0.0
2792480	Description: Running tcpdump on a bonding standby port resulted in the loss of the network.
	Keywords: NetDev
	Discovered in Release: 5.4-1.0.3.0
	Fixed in Release: 5.4-3.0.3.0
2696789	Description: Redesigning the locks around peer MR invalidation flow to avoid a potential deadlock as Peer-direct patch may cause deadlock due to lock inversion.
	Notes:
	<ul style="list-style-type: none"> For GPU drivers prior to r470, the user should update <code>nv_peer_mem</code> to the next version, probably 1.2. For GPU drivers from r470 or later branches shipped with <code>nvidia-peermem</code>, the driver will have an option to update to newer releases which take advantage of the redesigned MLNX_OFED support.
	Keywords: lock inversion, nv_peer_mem
	Discovered in Release: 5.4-1.0.3.0
2739689	Fixed in Release: 5.4-3.0.3.0
	Description: A race that resulted in a QCE with an error, caused errors in UMR QP. To prevent the UMR QP from getting into error, we fixed the MR deregistration flow (e.g., Peer lkey which is always revoked before destroying it).
	Keywords: QCE, UMR
	Discovered in Release: 5.4-1.0.3.0
2691656	Fixed in Release: 5.4-3.0.3.0
	Description: When using bonding, ibdev2netdev would sometimes match the infiniband device to the net device bonding interface, and sometimes to the underlying Infiniband net device interface. ibdev2netdev now skips InfiniBand net device bonding interfaces, and always matches InfiniBand devices to the underlying InfiniBand net device interfaces.
	Keywords: ibdev2netdev Bonding
	Discovered in Release: 5.0-1.0.0.0
2687643	Fixed in Release: 5.4-3.0.3.0
	Description: Fixed Decap flows inner IP_ECN match to take into account software modification of the match value according to RFC 6040 4.2.
	Keywords: decap, ASAP ² , ECN, RoCE
	Discovered in Release: 5.4-1.0.3.0
	Fixed in Release: 5.4-3.0.3.0

Internal Reference Number	Description
2691081	<p>Description: Removed metadata from the rpm package mlnx-ofa_kernel where it claimed to Provide an older version of rdma-core. This made sense in older versions where we needed to avoid installing rdma-core. But does not make sense anymore. And caused problems to some users installing rdma-core-devel through meta-packages.</p> <p>Keywords: Installation</p> <p>Discovered in Release: 5.4-1.0.3.0</p> <p>Fixed in Release: 5.4-3.0.3.0</p>
2727062	<p>Description: Removed manual build-time file list generation in mlnx-tools. Only keep it for python-installed files. And avoid guessing the version of python we use and the directory to which we install.</p> <p>Keywords: Installation</p> <p>Discovered in Release: 5.4-1.0.3.0</p> <p>Fixed in Release: 5.4-3.0.3.0</p>
2708220	<p>Description: Removed useless build-time editing of uninstall.sh in ofed-scripts that caused the build to fail (in the case of --add-kernel-support) in some rare cases.</p> <p>Keywords: Installation</p> <p>Discovered in Release: 5.4-1.0.3.0</p> <p>Fixed in Release: 5.4-3.0.3.0</p>
2730547	<p>Description: Some Dell OFED Factory Installation packages were missing dependencies. Removed the package rdma-core-devel from the Dell MLNX_OFED packages as it was not needed and some of its dependencies are not included.</p> <p>Keywords: Installation, Dell</p> <p>Discovered in Release: 5.4-1.0.3.0</p> <p>Fixed in Release: 5.4-3.0.3.0</p>
2699662	<p>Description: MLNX_OFED build scripts fixed to also build hcoll with CUDA support on RHEL8 x86_64 platforms.</p> <p>Keywords: Installation, CUDA</p> <p>Discovered in Release: 5.4-1.0.3.0</p> <p>Fixed in Release: 5.4-3.0.3.0</p>
2686877	<p>Description: Changing mtu takes too long. Reduced number of calls to synchronize_net to once for all channels.</p> <p>Keywords: mtu, synchronize_net</p> <p>Discovered in Release: 5.4-1.0.3.0</p> <p>Fixed in Release: 5.4-3.0.3.0</p>
2748328	<p>Description: When trying to upgrade a kmp package, it conflicts and needs user help to choose whether to replace it or not. The fix avoids conflicts from /usr/lib/rpm/kernel-module-subpackage script which was changed in the builder. Building the packages with kmp enabled on the other image will cause the issue to reproduce.</p> <p>Keywords: Upgrade, kmp Package</p> <p>Discovered in Release: 5.4-1.0.3.0</p>

Internal Reference Number	Description
	Fixed in Release: 5.4-3.0.3.0
2707023	Description: On Ubuntu and Debian systems for openvswitch-switch (in case installing using e.g. --ovs-dpdk or --with-openvswitch), the installer misses a run-time dependency of libpcap0.8.
	Keywords: Installation, Ubuntu, Debian
	Discovered in Release: 5.4-1.0.3.0
	Fixed in Release: 5.4-3.0.3.0
2563366	Description: The full path to the directory that contains the installer must not contain a space or any similar white-space character, otherwise the installer will fail.
	Keywords: Installation, White Space
	Discovered in Release: 5.3-1.0.0.1
	Fixed in Release: 5.4-3.0.3.0

Internal Reference Number	Description
2684302	Description: To support scalability, function representor channels were limited to 4. However in scenarios when SF are not used, certain use cases require representors to support a large number of channels. Hence, representor channel limit to 4 is applicable only when a PCI device, such as Scalable Function support, is enabled.
	Keywords: Representor Channels
	Discovered in Release: 5.3-1.0.0.1
	Fixed in Release: 5.4-1.0.3.0
2644217	Description: Matching on ipv4_ihl (internet header length) was supported only for outer headers. Support has been added for inner headers too.
	Keywords: Internet Header Length, ipv4_ihl
	Fixed in Release: 5.4-1.0.3.0
2626906	Description: When using one counter for both pop/push VLAN actions, the counter value is incorrect. Split the counter for pop_vlan_action_counter and push_vlan_action_counter.
	Keywords: Pop/Push VLAN
	Fixed in Release: 5.4-1.0.3.0
2653382	Description: Incorrect L3 decapsulation occurs when the original inner frame is small and was padded to comply with minimum frame size of 64-bytes.
	Keywords: SW Steering, Decapsulation, Padding

Internal Reference Number	Description
	Fixed in Release: 5.4-1.0.3.0
2612725	Description: dapl and libmlx4 are needed by libdat2 and libdpdk. In order to remove or update dapl, its dependencies need to be removed.
	Keywords: dapl, libmlx4
	Discovered in Release: 5.2-1.0.4.0
	Fixed in Release: 5.4-1.0.3.0
2649134	Description: An override of log_max_qp by other devices occurs if the devices share the same mlx5_core module.
	Keywords: log_max_qp, mlx5_core
	Fixed in Release: 5.4-1.0.3.0
2638029	Description: A synchronization issue where closing and opening channels (which may happen on configuration changes such as changing number of channels) may cause null pointer dereference in function mlx5e_select_queue.
	Keywords: mlx5e_select_queue, Synchronization, Tx
	Fixed in Release: 5.4-1.0.3.0
2678982	Description: Enabling tx-udp_tnl-csum-segmentation has no effect on the driver. tx-udp_tnl-csum-segmentation has been moved to "off [fixed]".
	Keywords: tx-udp_tnl-csum-segmentation
	Discovered in Release: 5.4-0.5.1.1
	Fixed in Release: 5.4-1.0.3.0
2610870	Description: Some MLNX_OFED dkms packages ignored (install-time) build errors and considered the packages properly built. Those errors are now not ignored and indicated as package installation errors.
	Keywords: dkms, Installation
	Discovered in Release: 5.2-2.2.0.0
	Fixed in Release: 5.4-1.0.3.0
2617820	Description: Old udevd versions could get stuck renaming network devices, leaving interfaces named eth* instead of enp*. Updating the systemd version resolves this issue. For example, if an issue detected on RHEL 7.6 with systemd-219-62, updating the systemd version to systemd-219-67 resolves the issue.
	Keywords: udev, systemd, RHEL
	Discovered in Release: 5.4-0.5.1.1
	Fixed in Release: 5.4-1.0.3.0
2632768	Description: Flows with t commit action with ct state -trk are not be offloaded (i.e., table=0,ct_state=-trk,ip actions=ct(commit,table=1)).
	Keywords: ASAP ² , Connection Tracking
	Fixed in Release: 5.4-1.0.3.0
2247143	Description: Connection tracking over VF LAG with tunnel encapsulation/decapsulation is not supported and may cause traffic drop.

Internal Reference Number	Description
	<p>Keywords: ASAP², Connection Tracking, VF LAG, Tunnel</p> <p>Discovered in Release: 5.0-1.0.0.0</p> <p>Fixed in Release: 5.4-1.0.3.0</p>
2597327	<p>Description: When stack size is limit to 1024, OFED compilation fails.</p> <p>Keywords: Compilation, Stack</p> <p>Discovered in Release: 5.3-1.0.0.1</p> <p>Fixed in Release: 5.4-1.0.3.0</p>
2609641	<p>Description: Setting rate/burst values higher than 2,147,483,648 are rejected.</p> <p>Keywords: VF Metering</p> <p>Discovered in Release: 5.3-1.0.0.1</p> <p>Fixed in Release: 5.4-1.0.3.0</p>
2626920	<p>Description: Offloaded remote mirroring flows on tunnel device caused forwarded traffic to VF to not be decapsulated.</p> <p>Keywords: ASAP², Offload, Remote Mirroring, Tunnel</p> <p>Discovered in Release: 5.3-1.0.0.1</p> <p>Fixed in Release: 5.4-1.0.3.0</p>
2660247	<p>Description: Trying to set VPort match mode on VF (cat/sys/class/net/enp8s0f2/compat/devlink/vport_match_mode), leads to kernel crash.</p> <p>Keywords: ASAP², Kernel Crash</p> <p>Discovered in Release: 5.3-1.0.0.1</p> <p>Fixed in Release: 5.4-1.0.3.0</p>
2667484	<p>Description: OVS flows are not being offloaded over socket-direct devices.</p> <p>Keywords: ASAP², Socket-Direct</p> <p>Discovered in Release: 5.3-1.0.0.1</p> <p>Fixed in Release: 5.4-1.0.3.0</p>
2663042	<p>Description: When VXLAN is configured and illegal route is added, the system crashes with call trace.</p> <p>Keywords: ASAP², Offload, Tunnel, Call Trace</p> <p>Discovered in Release: 5.3-1.0.0.1</p> <p>Fixed in Release: 5.4-1.0.3.0</p>
2354761	<p>Description: If any traffic is sent before the netdev goes up for the first time, a division by zero caused by a modulo operation may occur in ndo_select_queue, leading to a kernel panic.</p> <p>Keywords: NetDev; ndo_select_queue</p> <p>Discovered in Release: 5.3-1.0.0.1</p> <p>Fixed in Release: 5.4-1.0.3.0</p>

Internal Reference Number	Description
2562053/2667551	Description: After restarting driver, the x86 host may be in grace period and may not recover on its own. As part of the fix, 5 FW_fatal recoveries are allowed within the 20-minute grace period. As a result, the grace period in the devlink health show command will appear as 0 for FW_fatal reporter.
	Keywords: BlueField Reload, recovery, reset flow
	Discovered in Release: 5.2-2.2.0.0
	Fixed in Release: 5.4-1.0.3.0

Internal Reference Number	Issue
2393352	Description: Using "--with-openswitch" flag during MLNX_EN installation may not work on Debian 10 systems.
	Keywords: --with-openswitch, Debian
	Discovered in Release: 5.2-1.0.4.0
	Fixed in Release: 5.3-1.0.0.1
2445058	Description: ib_uverbs module parameter disable_raw_qp_enforcement is deprecated and should no longer be used.
	Keywords: disable_raw_qp_enforcement, ib_uverbs
	Discovered in Release: 5.2-2.2.0.0
	Fixed in Release: 5.3-1.0.0.1
2434650	Description: Fixed an issue in ConnectX-5 and earlier that when the module is missing, the driver reported a connector type that is different than OTHER.
	Keywords: Module, Connector Type
	Discovered in Release: 5.2-2.2.0.0
	Fixed in Release: 5.3-1.0.0.1
2434650	Description: Solved a compilation error by fixing a backport issue with unpin_user_pages_dirty_lock function.
	Keywords: Memory, unpin_user_pages_dirty_lock
	Discovered in Release: 5.2-2.2.0.0
	Fixed in Release: 5.3-1.0.0.1
2505615	Description: Fixed an issue where VLAN header was not popped on VF Rx when the eSwitch priority tagging was configured.
	Keywords: ASAP ² , Priority Tagging, VLAN
	Discovered in Release: 5.2-2.2.0.0
	Fixed in Release: 5.3-1.0.0.1
2494257	Description: Fixed connection tracking (CT) offload in NIC mode by using correct steering domain for the rules.
	Keywords: ASAP ² , Connection Tracking, NIC Mode

Internal Reference Number	Issue
	Discovered in Release: 5.2-2.2.0.0
	Fixed in Release: 5.3-1.0.0.1
2461213	Description: Fixed an issue where offload of rules from OVS internal port to uplink failed.
	Keywords: ASAP ² , OVS
	Discovered in Release: 5.2-2.2.0.0
	Fixed in Release: 5.3-1.0.0.1
2444523	Description: Fixed an issue in the tunnel mishandling that can happen when the tunnel overlay device is an OVS internal port.
	Keywords: ASAP ² , OVS internal port offloading
	Discovered in Release: 5.2-2.2.0.0
	Fixed in Release: 5.3-1.0.0.1
2566354	Description: Fixed incorrect parsing of network configuration when the option --net (-n) was given to mlnxofedinstall: get network configuration from the output of 'ip' instead of 'ifconfig'.
	Keywords: Installation
	Fixed in Release: 5.3-1.0.0.1
2495065	Description: Dropped unsupported devices from OFED rdma-core description.
	Keywords: rdma-core
	Discovered in Release: 5.2-2.2.0.0
	Fixed in Release: 5.3-1.0.0.1
2482696	Description: Backported MLNX_EN kernel to support elrepo 5.8 kernel.
	Keywords: add-kernel-support
	Discovered in Release: 5.2-1.0.4.0
	Fixed in Release: 5.3-1.0.0.1
2481104	Description: Fixed ability to build xpmem on kernel version 5.6.
	Keywords: add-kernel-support, xpmem
	Discovered in Release: 5.2-1.0.4.0
	Fixed in Release: 5.3-1.0.0.1
2440062	Description: Fixed an issue where kernel build on SLES 15 systems that configures scripts assume SLES 15 systems have /etc/SuSE-release or /etc/SUSE-brand. These files no longer exist on SLES 15.
	Keywords: add-kernel-support, SLES 15
	Discovered in Release: 5.2-1.0.4.0
	Fixed in Release: 5.3-1.0.0.1
2445146	Description: Fixed an issue where running data on Geneve tunnel on a VF may result in CQE error and a failure to transmit data.
	Keywords: Virtual Function

Internal Reference Number	Issue
	<p>Discovered in Release: 5.2-1.0.4.0</p> <p>Fixed in Release: 5.3-1.0.0.1</p>
2494008	<p>Description: Fixed an issue where the driver silently ignores the settings of an already-set ECN value (0->0, 1->1) via sysfs.</p> <p>Keywords: RDMA, ECN</p> <p>Discovered in Release: 5.2-1.0.4.0</p> <p>Fixed in Release: 5.3-1.0.0.1</p>
2581127	<p>Description: Fixed an issue where KVS offload, under certain conditions, takes too long. Improved malloc performance by increasing the memory reuse and reducing the stress on malloc and free.</p> <p>Keywords: MLNX5DR, Software Steering</p> <p>Discovered in Release: 5.2-1.0.4.0</p> <p>Fixed in Release: 5.3-1.0.0.1</p>
2502564	<p>Description: Fixed an issue where when using switchdev mode with SMFS, inserting duplicate rules from userspace was not supported (required when there are a few instances of the same application). As part of the fix, added support for update_fte which is called in case a duplicate rule is being added.</p> <p>Keywords: SwitchDev, Steering</p> <p>Fixed in Release: 5.3-1.0.0.1</p>
2433351	<p>Description: Fixed an issue where creating 127 ports on each VF may fail as the current kernel does not support an RDMA device with more than 255 ports.</p> <p>Keywords: VF, RDMA, virtualization</p> <p>Discovered in Release: 5.2-2.2.0.0</p> <p>Fixed in Release: 5.3-1.0.0.1</p>
2333971	<p>Description: Fixed an issue where changing the "other" channels count by "ethtool -L <interface> other <count>" command on Kernel 5.10 may cause a kernel panic.</p> <p>Keywords: Kernel 5.10, kernel panic, ethtool, "other" channels</p> <p>Discovered in Release: 5.2-1.0.4.0</p> <p>Fixed in Release: 5.3-1.0.0.1</p>
1731939	<p>Description: Get/Set Forward Error Correction FEC configuration is not supported on ConnectX-6 HCAs with 200GbE speed rate.</p> <p>Workaround: N/A</p> <p>Discovered in Release: 4.6-1.0.1.1</p> <p>Fixed in Release: 5.3-1.0.0.1</p>
2454952	<p>Description: Fixed an issue where MLNX_EN cannot be built on top of Kernel 5.4.87.</p> <p>Workaround: operating system, kernel</p> <p>Discovered in Release: 5.2-2.2.0.0</p> <p>Fixed in Release: 5.3-1.0.0.1</p>

Internal Reference Number	Issue
2383355	<p>Description: Fixed an issue where Switch and eSwitch offloads are not supported for SR-IOV and its sub functions when installing MLNX_EN over upstream kernel v5.10 or higher.</p> <p>Keywords: eSwitch, Kernel, SR-IOV</p> <p>Discovered in Release: 5.2-1.0.4.0</p> <p>Fixed in Release: 5.3-1.0.0.1</p>
2083942	<p>Description: Fixed the issue where the content of file /sys/class/net/<NETIF>/statistics/multicast may have been out of date and may have displayed values lower than the real values.</p> <p>Keywords: Multicast counters</p> <p>Discovered in Release: 5.0-1.0.0.0</p> <p>Fixed in Release: 5.2-1.0.4.0</p>
2282316	<p>Description: Fixed the issue where ERSPAN protocol was available only when turning off Tx checksum offload.</p> <p>Keywords: ERSPAN, TX checksum offload</p> <p>Discovered in Release: 5.1-2.5.8.0</p> <p>Fixed in Release: 5.2-1.0.4.0</p>
2310695	<p>Description: Fixed a udev script issue which caused non-NVIDIA devices to be renamed.</p> <p>Keywords: udev, naming</p> <p>Discovered in Release: 5.0-1.0.0.0</p> <p>Fixed in Release: 5.2-1.0.4.0</p>
2334518	<p>Description: Fixed missing representor statistics when using ifconfig.</p> <p>Keywords: SwitchDev, representor, statistics, ifconfig</p> <p>Discovered in Release: 5.1-2.5.8.0</p> <p>Fixed in Release: 5.2-1.0.4.0</p>
2342348	<p>Description: Fixed wrong value of skb mark of received packets on representors.</p> <p>Keywords: SwitchDev, skb mark</p> <p>Discovered in Release: 5.1-2.5.8.0</p> <p>Fixed in Release: 5.2-1.0.4.0</p>
2363982	<p>Description: Fixed an issue which caused second port representors to be named as first port representors.</p> <p>Keywords: SwitchDev, udev, representor</p> <p>Discovered in Release: 5.1-2.5.8.0</p> <p>Fixed in Release: 5.2-1.0.4.0</p>
2020260	<p>Description: Fixed the issue of when changing the Trust mode to DSCP, there was an interval between the change taking effect in the hardware and updating the inline mode of the SQ in the driver. If any traffic was transmitted during this interval, the driver would not inline enough headers, resulting in a CQE error in the NIC.</p> <p>Keywords: DSCP, inline, SQ, CQE</p>

Internal Reference Number	Issue
	Discovered in Release: 5.0-1.0.0.0
	Discovered in Release: 5.1-1.0.4.0
2105631	Description: Removed IBV_FLOW_ATTR_FLAGS_ALLOW_LOOP_BACK flag as it is not used by the kernel.
	Keywords: IBV_FLOW_ATTR_FLAGS_ALLOW_LOOP_BACK
	Discovered in Release: 4.7-1.0.0.1
	Fixed in Release: 5.1-1.0.4.0
2099043	Description: Added QP isolation to improve SW steering performance under high packet load. This will allow SW steering RC QP to be executed on a separate scheduling queue without competing over hardware resources.
	Keywords: Software steering, ASAP, connection tracking, CT
	Discovered in Release: 5.0-1.0.0.0
	Fixed in Release: 5.1-1.0.4.0
2097045	Description: Userspace Software Steering using mlx5dv_dr API support on ConnectX-6 Dx adapter cards is now at GA level.
	Keywords: Software Steering, SW, mlx5dv_dr, ConnectX-6 Dx
	Discovered in Release: 5.0-1.0.0.0
	Fixed in Release: 5.1-1.0.4.0
2132332	Description: Fixed a sporadic reporting bandwidth issue in case of running with <code>--run_indefinitely</code> flag.
	Keywords: perftest, bandwidth
	Discovered in Release: 5.0-1.0.0.0
	Fixed in Release: 5.1-1.0.4.0
2151658	Description: Optimized XRC target lookup by modifying the locking scheme to enable multiple readers and changing the linked list that holds the QPs to xarray.
	Keywords: XRC, QP, xarray
	Discovered in Release: 5.0-1.0.0.0
	Fixed in Release: 5.1-1.0.4.0
2196118	Description: Fixed a driver issue that led to panic after DPDK application crashes.
	Keywords: DPDK, panic
	Discovered in Release: 5.0-1.0.0.0
	Fixed in Release: 5.1-1.0.4.0
2245228	Description: Fixed an issue of a crash when attempting to access roce_enable sysfs in unprobed VFs.
	Keywords: roce_enable, unprobed VFs
	Discovered in Release: 5.0-1.0.0.0
	Fixed in Release: 5.1-1.0.4.0

Internal Reference Number	Issue
2061294	<p>Description: Fixed a race of commands executed by command interface in parallel to AER recovery causing the kernel to crash.</p> <p>Keywords: mlx5e, AER</p> <p>Discovered in Release: 4.7-1.0.0.1</p> <p>Fixed in Release: 5.1-1.0.4.0</p>
1731005	<p>Description: Regenerated package repository in the correct location after rebuilding the kernel using <code>add-kernel-support</code>. This allows for installing the newly generated packages with a package manager.</p> <p>Keywords: add-kernel-support, RPM, deb</p> <p>Discovered in Release: 5.0-1.0.0.0</p> <p>Fixed in Release: 5.1-1.0.4.0</p>
2172130	<p>Description: Fixed an issue with metadata packages generation in the eth-only directory. This allows using the directory as a repository for package managers.</p> <p>Keywords: Metadata packages</p> <p>Discovered in Release: 5.0-1.0.0.0</p> <p>Fixed in Release: 5.1-1.0.4.0</p>
2214543	<p>Description: Moved <code>ibdev2netdev</code> script from <code>/usr/bin</code> to <code>/usr/sbin</code> in the RPM package to avoid package conflict with RHEL 8 and consequent <code>MLNX_EN</code> installation failure on some systems.</p> <p>Keywords: <code>ibdev2netdev</code>, RPM, RHEL, RedHat</p> <p>Discovered in Release: 5.0-1.0.0.0</p> <p>Fixed in Release: 5.1-1.0.4.0</p>
2211311	<p>Description: Fixed an issue where Rx port buffers cell size was wrong, leading to wrong buffers size reported by <code>mlnx_qos/netdev qos/buffer_size sysfs</code>.</p> <p>Keywords: mlx5e, RX buffers, <code>mlnx_qos</code></p> <p>Discovered in Release: 5.0-1.0.0.0</p> <p>Fixed in Release: 5.1-1.0.4.0</p>
2111349	<p>Description: Fixed the issue where <code>ethtool --show-fec/ --get-fec</code> were not supported over ConnectX-6 and ConnectX-6 Dx adapter cards.</p> <p>Keywords: Ethtool, ConnectX-6 Dx</p> <p>Discovered in Release: 5.0-1.0.0.0</p> <p>Fixed in Release: 5.1-1.0.4.0</p>
2165668	<p>Description: Fixed an issue related to mlx5 command interface that in some scenarios caused the driver to hang.</p> <p>Keywords: ConnectX-5, mlx5, panic</p> <p>Discovered in Release: 4.6-1.0.1.1</p> <p>Fixed in Release: 5.1-1.0.4.0</p>

Internal Reference Number	Issue
2119984	<p>Description: Fixed the issue where IPsec crypto offloads did not work when ESN was enabled.</p> <p>Keywords: IPsec, ESN</p> <p>Discovered in Release: 5.0-1.0.0.0</p> <p>Fixed in Release: 5.1-1.0.4.0</p>
1630228	<p>Description: Fixed the issue where tunnel stateless offloads were wrongly forbidden for E-Switch manager function.</p> <p>Keywords: Stateless offloads cap</p> <p>Discovered in Release: 4.6-1.0.1.1</p> <p>Fixed in Release: 5.1-1.0.4.0</p>
2089996	<p>Description: Fixed the issue where dump flows were not supported and may have been corrupted when using tc tool with connection tracking rules.</p> <p>Keywords: ASAP, iproute2, tc, connection tracking</p> <p>Discovered in Release: 5.0-1.0.0.0</p> <p>Fixed in Release: 5.1-1.0.4.0</p>
2094216	<p>Description: Fixed the issue of when one of the LAG slaves went down, LAG deactivation failed, ultimately causing bandwidth degradation.</p> <p>Keywords: RoCE LAG</p> <p>Discovered in Release: 4.7-3.2.9.0</p> <p>Fixed in Release: 5.1-1.0.4.0</p>
2133778	<p>Description: The mlx5 driver maintains a subdirectory for every open eth port in /sys/kernel/debug/. For the default network namespace, the sub-directory name is the name of the interface, like "eth8". The new convention for the network interfaces moved to the non-default network namespaces is the interfaces name followed by "@" and the port's PCI ID. For example: "eth8@0000:af:00.3".</p> <p>Keywords: Namespace</p> <p>Discovered in Release: 4.7-3.2.9.0</p> <p>Fixed in Release: 5.1-1.0.4.0</p>
2076546	<p>Description: Fixed the issue where in RPM-based OSs with non-default kernels, using repositories after re-creating the installer (using --add-kernel-support) would result in improper installation of the drivers.</p> <p>Keywords: Installation, OS</p> <p>Discovered in Release: 4.7-1.0.0.1</p> <p>Fixed in Release: 5.1-1.0.4.0</p>
2114957	<p>Description: Fixed the issue where MLNX_EN installation may have depended on python2 package even when attempting to install it on OSs whose default package is python3.</p> <p>Keywords: Installation, python</p> <p>Discovered in Release: 4.7-3.2.9.0</p> <p>Fixed in Release: 5.1-1.0.4.0</p>

Internal Reference Number	Issue
2122684	<p>Description: Fixed the issue where OFED uninstallation resulted in the removal of dependency packages, such as qemu-system-* (qemu-system-x86).</p> <p>Keywords: Uninstallation, dependency, qemu-system-x86</p> <p>Discovered in Release: 4.7-3.2.9.0</p> <p>Fixed in Release: 5.1-1.0.4.0</p>
2135476	<p>Description: Added KMP ability to install MLNX_EN Kernel modules on SLES12 SP5 and SLES15 kernel maintenance updates.</p> <p>Keywords: KMP, SLES, kernel</p> <p>Discovered in Release: 4.7-3.2.9.0</p> <p>Fixed in Release: 5.1-1.0.4.0</p>
2143258	<p>Description: Fixed a typo in perftest package where help messages wrongly displayed the conversion result between Gb/s and MB/s (20^2 instead of 2^20).</p> <p>Keywords: perftest</p> <p>Discovered in Release: 4.7-3.2.9.0</p> <p>Fixed in Release: 5.1-1.0.4.0</p>
2149577	<p>Description: Fixed the issue where openibd script load used to fail when esp6_offload module did not load successfully.</p> <p>Keywords: openibd, esp6_offload</p> <p>Discovered in Release: 4.7-3.2.9.0</p> <p>Fixed in Release: 5.1-1.0.4.0</p>
2163879	<p>Description: Added dependency of package mpi-selectors on perl-Getopt-Long system package. On minimal installs of RPM-based OSs, installing mpi-selectors will also install the required system package perl-Getopt-Long.</p> <p>Keywords: Dependency, perl-Getopt-Long</p> <p>Discovered in Release: 4.7-3.2.9.0</p> <p>Fixed in Release: 5.1-1.0.4.0</p>
2119017	<p>Description: Fixed the issue where injecting EEH may cause extra Kernel prints, such as: "EEH: Might be infinite loop in mlx5_core driver".</p> <p>Keywords: EEH, kernel</p> <p>Discovered in Release: 4.7-3.2.9.0</p> <p>Fixed in Release: 5.1-1.0.4.0</p>
2107532	<p>Description: Fixed the issue where in certain rare scenarios, due to Rx page not being replenished, the same page fragment mistakenly became assigned to two different Rx descriptors.</p> <p>Keywords: Memory corruption, Rx page recycle</p> <p>Discovered in Release: 4.7-3.2.9.0</p> <p>Fixed in Release: 5.1-1.0.4.0</p>

Internal Reference Number	Issue
2116234	Description: Fixed the issue where ibsim was missing after OFED installation.
	Keywords: ibsim, installation
	Discovered in Release: 5.0-1.0.0.0
	Fixed in Release: 5.1-1.0.4.0
2116233	Description: Fixed an issue where ucx-kmem was missing after OFED installation.
	Keywords: ucx-kmem, installation
	Discovered in Release: 5.0-1.0.0.0
	Fixed in Release: 5.1-1.0.4.0
2109716	Description: Fixed a dependency issue between systemd and RDMA-Core.
	Keywords: Dependency, RDMA-Core
	Discovered in Release: 5.0-1.0.0.0
	Fixed in Release: 5.1-1.0.4.0
2107776	Description: Fixed a driver load issue with Errata-kernel on SLES15 SP1.
	Keywords: Load, SLES, Errata
	Discovered in Release: 5.0-1.0.0.0
	Fixed in Release: 5.1-1.0.4.0
2105536	Description: Fixed an issue in the Hairpin feature which prevented adding hairpin flows using TC tool.
	Keywords: Hairpin, TC
	Discovered in Release: 5.0-1.0.0.0
	Fixed in Release: 5.1-1.0.4.0
2090321	Description: Fixed the issue where WQ queue flushing was not handled properly in the event of EEH.
	Keywords: WQ, EEH
	Discovered in Release: 4.7-1.0.0.1
	Fixed in Release: 5.1-1.0.4.0
2076311	Description: Fixed a rare kernel crash scenario when exiting an application that uses RMPP mads intensively.
	Keywords: MAD RMPP
	Discovered in Release: 4.0-1.0.1.0
	Fixed in Release: 5.1-1.0.4.0
2094545	Description: Fixed the issue where perftest applications (ib_read_*, ib_write_* and others) supplied with MLNX_EN v5.0 and above did not work correctly if corresponding applications on another side of client-server communication were supplied with previous versions of MLNX_EN due to an interoperability issue.
	Keywords: perftest, interoperability
	Discovered in Release: 5.0-1.0.0.0

Internal Reference Number	Issue
	Fixed in Release: 5.1-1.0.4.0
2096998	Description: Fixed the issue where NEO-Host could not be installed from the MLNX_EN package when working on Ubuntu and Debian OSs.
	Keywords: NEO-Host, Ubuntu, Debian
	Discovered in Release: 5.0-1.0.0.0
	Fixed in Release: 5.1-1.0.4.0
2094012	Description: Fixed the issue where MLNX_EN installation failed to upgrade firmware version on ConnectX-6 Dx NICs with secure-fw.
	Keywords: ConnectX-6 Dx, installation, firmware, NIC
	Discovered in Release: 5.0-1.0.0.0
	Fixed in Release: 5.1-1.0.4.0
2057076	Description: Added support for installing MLNX_EN using <code>--add-kernel-support</code> option over RHEL 8 OSs.
	Keywords: --add-kernel-support, installation, RHEL
	Discovered in Release: 5.0-1.0.0.0
	Fixed in Release: 5.1-1.0.4.0
2090186	Description: Fixed a possible kernel crash scenario when AER/slot reset is done in parallel to user space commands execution.
	Keywords: mlx5_core, AER, slot reset
	Discovered in Release: 4.3-1.0.1.0
	Fixed in Release: 5.1-1.0.4.0
2093410	Description: Added missing ECN configuration under sysfs for PFs in SwitchDev mode.
	Keywords: sysfs, ASAP, SwitchDev, ECN
	Discovered in Release: 4.7-3.2.9.0
	Fixed in Release: 5.1-1.0.4.0
2036394	Description: Added driver support for kernels with the old XDP_REDIRECT infrastructure that uses the following NetDev operations: <code>.ndo_xdp_flush</code> and <code>.ndo_xdp_xmit</code> .
	Keywords: XDP_REDIRECT, Soft lockup
	Discovered in Release: 4.7-3.2.9.0
	Fixed in Release: 5.0-1.0.0.0
2072871	Description: Fixed an issue where the usage of <code>--excludedocs</code> Open MPI RPM option resulted in the removal of non-documentation related files.
	Keywords: --excludedocs, Open MPI, RPM
	Discovered in Release: 4.5-1.0.1.0
	Fixed in Release: 5.0-1.0.0.0
2060216	Description: Legacy mlnx-libs are now installed by default on SLES11 SP3 OS, as building MLNX_EN on RDMA-Core based packages with this OS is not supported.

Internal Reference Number	Issue
	<p>Keywords: mlnx-libs, SLES, RDMA-Core</p> <p>Discovered in Release: 4.7-3.2.9.0</p> <p>Fixed in Release: 5.0-1.0.0.0</p>
2072884	<p>Description: Removed all cases of automated loading of MLNX_EN kernel modules outside of openibd to preserve the startup process of previous MLNX_EN versions. These loads conflict with openibd, which has its own logic to overcome issues. Such issues can be inbox driver load instead of MLNX_EN, or module load with wrong parameter value. They might also load modules while openibd is trying to unload the driver stack.</p> <p>Keywords: Installation, openibd, RDMA-Core</p> <p>Discovered in Release: 4.7-3.2.9.0</p> <p>Fixed in Release: 5.0-1.0.0.0</p>
2052037	<p>Description: Disabled automated loading of some modules through udev triggers to preserve the startup process of previous MLNX_EN versions.</p> <p>Keywords: Installation, udev, RDMA-Core</p> <p>Discovered in Release: 4.7-3.2.9.0</p> <p>Fixed in Release: 5.0-1.0.0.0</p>
2022634	<p>Description: Fixed a typo in the packages build command line which could cause the installation of MLNX_EN on SLES OSs to fail when using the option --without-depcheck.</p> <p>Keywords: Installation, SLES</p> <p>Discovered in Release: 4.7-3.2.9.0</p> <p>Fixed in Release: 5.0-1.0.0.0</p>
2022619	<p>Description: Fixed the issue where uninstallation of MLNX_EN would hang due to a bug in the package dependency check.</p> <p>Keywords: Uninstallation, dependency</p> <p>Discovered in Release: 4.7-1.0.0.1</p> <p>Fixed in Release: 5.0-1.0.0.0</p>
1995843	<p>Description: ibdump is now provided with the default rdma-core-based build.</p> <p>Keywords: ibdump, RDMA-Core</p> <p>Discovered in Release: 4.7-3.2.9.0</p> <p>Fixed in Release: 5.0-1.0.0.0</p>
1995631	<p>Description: Proper package dependencies are now set on Debian and Ubuntu libibverbs-dev package that is generated from RDMA-Core.</p> <p>Keywords: Dependency, libibverbs, RDMA-Core</p> <p>Discovered in Release: 4.7-3.2.9.0</p> <p>Fixed in Release: 5.0-1.0.0.0</p>
2047221	<p>Description: Reference count (refcount) for RDMA connection ID (cm_id) was not incremented in rdma_resolve_addr() function, resulting in a cm_id use-after-free access. A fix was applied to increment the cm_id refcount.</p>

Internal Reference Number	Issue
	<p>Keywords: rdma_resolve_addr(), cm_id</p> <p>Discovered in Release: 4.6-1.0.1.1</p> <p>Fixed in Release: 5.0-1.0.0.0</p>
2045181	<p>Description: Fixed a race condition which caused kernel panic when moving two ports to SwitchDev mode at the same time.</p> <p>Keywords: ASAP, SwitchDev, race</p> <p>Discovered in Release: 4.7-1.0.0.1</p> <p>Fixed in Release: 5.0-1.0.0.0</p>
2004488	<p>Description: Allowed accessing sysfs hardware counters in SwitchDev mode.</p> <p>Keywords: ASAP, hardware counters, sysfs, SwitchDev</p> <p>Discovered in Release: 4.7-1.0.0.1</p> <p>Fixed in Release: 5.0-1.0.0.0</p>
2030943	<p>Description: Function smp_processor_id() is called in the RX page recycle flow to determine the core to run on. This is intended to run in NAPI context. However, due to a bug in backporting, the RX page recycle was mistakenly called also in the RQ close flow when not needed.</p> <p>Keywords: Rx page recycle, smp_processor_id</p> <p>Discovered in Release: 4.6-1.0.1.1</p> <p>Fixed in Release: 5.0-1.0.0.0</p>
2074487	<p>Description: Fixed an issue where port link state was automatically changed (without admin state involvement) to "UP" after reboot.</p> <p>Keywords: Link state, UP</p> <p>Discovered in Release: 4.7-1.0.0.1</p> <p>Fixed in Release: 5.0-1.0.0.0</p>
2064711	<p>Description: Fixed an issue where RDMA CM connection failed when port space was small.</p> <p>Keywords: RDMA CM</p> <p>Discovered in Release: 4.7-1.0.0.1</p> <p>Fixed in Release: 5.0-1.0.0.0</p>
2076424	<p>Description: Traffic mirroring with OVS offload and non-offload over VxLAN interface is now supported. Note: For kernel 4.9, make sure to use a dedicated OVS version.</p> <p>Keywords: VxLAN, OVS</p> <p>Discovered in Release: 4.7-3.2.9.0</p> <p>Fixed in Release: 5.0-1.0.0.0</p>
1828321	<p>Description: Fixed the issue of when working with VF LAG while the bond device is in active-active mode, running fwreset would result in unequal traffic on both PFs, and PFs would not reach line rate.</p> <p>Keywords: VF LAG, bonding, PF</p>

Internal Reference Number	Issue
	<p>Discovered in Release: 4.6-1.0.1.1</p> <p>Fixed in Release: 5.0-1.0.0.0</p>
1975293	<p>Description: Installing OFED with <code>--with-openvswitch</code> flag no longer requires manual removal of the existing Open vSwitch.</p> <p>Keywords: OVS, Open vSwitch, openvswitch</p> <p>Discovered in Release: 4.7-3.2.9.0</p> <p>Fixed in Release: 5.0-1.0.0.0</p>
1939719	<p>Description: Fixed an issue of when running openibd restart after the installation of MLNX_EN on SLES12 SP5 and SLES15 SP1 OSs with the latest Kernel (v4.12.14) resulted in an error that the modules did not belong to that Kernel. This was due to the fact that the module installed by MLNX_EN was incompatible with new Kernel's module.</p> <p>Keywords: SLES, operating system, OS, installation, Kernel, module</p> <p>Discovered in Release: 4.7-3.2.9.0</p> <p>Fixed in Release: 5.0-1.0.0.0</p>
2001966	<p>Description: Fixed an issue of when bond was created over VF netdevices in SwitchDev mode, the VF netdevice would be treated as representor netdevice. This caused the mlx5_core driver to crash in case it received netdevice events related to bond device.</p> <p>Keywords: PF, VF, SwitchDev, netdevice, bonding</p> <p>Discovered in Release: 4.7-3.2.9.0</p> <p>Fixed in Release: 5.0-1.0.0.0</p>
1816629	<p>Description: Fixed an issue where following a bad affinity occurrence in VF LAG mode, traffic was sent after the port went up/down in the switch.</p> <p>Keywords: Traffic, VF LAG</p> <p>Discovered in Release: 4.6-1.0.1.1</p> <p>Fixed in Release: 5.0-1.0.0.0</p>
1718531	<p>Description: Added support for VLAN header rewrite on CentOS 7.2 OS.</p> <p>Keywords: VLAN, ASAP, switchdev, CentOS 7.2</p> <p>Discovered in Release: 4.6-1.0.1.1</p> <p>Fixed in Release: 5.0-1.0.0.0</p>
1556337	<p>Description: Fixed the issue where adding VxLAN decapsulation rule with enc_tos and enc_ttl failed.</p> <p>Keywords: VxLAN, decapsulation</p> <p>Discovered in Release: 4.7-1.0.0.1</p> <p>Fixed in Release: 5.0-1.0.0.0</p>
1921799	<p>Description: Fixed the issue where MLNX_EN installation over SLES15 SP1 ARM OSs failed unless <code>--add-kernel-support</code> flag was added to the installation command.</p> <p>Keywords: SLES, installation</p> <p>Discovered in Release: 4.7-1.0.0.1</p>

Internal Reference Number	Issue
	Fixed in Release: 4.7-3.2.9.0
1973828	Description: Fixed wrong EEPROM length for small form factor (SFF) 8472 from 256 to 512 bytes.
	Keywords: EEPROM, SFF
	Discovered in Release: 4.7-1.0.0.1
	Fixed in Release: 4.7-3.2.9.0
1915553	Description: Fixed the issue where errno field was not sent in all error flows of ibv_reg_mr API.
	Keywords: ibv_reg_mr
	Discovered in Release: 4.6-1.0.1.1
	Fixed in Release: 4.7-3.2.9.0
1970901	Description: Fixed the issue where mlx5 IRQ name did not change to express the state of the interface.
	Keywords: Ethernet, PCIe, IRQ
	Discovered in Release: 4.7-1.0.0.1
	Fixed in Release: 4.7-3.2.9.0
1915587	Description: Udaddy application is now functional in Legacy mode.
	Keywords: Udaddy, MLNX_EN legacy, RDMA-CM
	Discovered in Release: 4.7-1.0.0.1
	Fixed in Release: 4.7-3.2.9.0
1931421	Description: Added support for E-Switch (SR-IOV Legacy) mode in RHEL 7.7 OSs.
	Keywords: E-Switch, SR-IOV, RHEL, RedHat
	Discovered in Release: 4.7-1.0.0.1
	Fixed in Release: 4.7-3.2.9.0
1945411/1839353	Description: Fixed the issue of when XDP_REDIRECT fails, pages got double-freed due to a bug in the refcnt_bias feature.
	Keywords: XDP, XDP_REDIRECT, refcnt_bias
	Discovered in Release: 4.6-1.0.1.1
	Fixed in Release: 4.7-3.2.9.0
1976482	Description: Added support for enabling SwitchDev mode in MLNX_EN.
	Keywords: SwitchDev
	Discovered in Release: 4.7-1.0.0.1
	Fixed in Release: 4.7-3.2.9.0
1734102	Description: Fixed the issue where Ubuntu v16.04.05 and v16.04.05 OSs could not be used with their native kernels.
	Keywords: Ubuntu, Kernel, OS
	Discovered in Release: 4.6-1.0.1.1

Internal Reference Number	Issue
1758983	<p>Fixed in Release: 4.7-1.0.0.1</p> <p>Description: Installing MLNX_EN on RHEL 7.6 OSs platform x86_64 and RHEL 7.6 ALT OSs platform PPCLE using YUM is now supported.</p> <p>Keywords: RHEL, RedHat, YUM, OS, operating system</p> <p>Discovered in Release: 4.6-1.0.1.1</p> <p>Fixed in Release: 4.7-1.0.0.1</p>
1800525	<p>Description: When configuring the Time-stamping feature, CQE compression will be disabled. This fix entails the removal of a warning message that appeared upon attempting to disable CQE compression when it has already been disabled.</p> <p>Keywords: Time-stamping, CQE compression</p> <p>Discovered in Release: 4.6-1.0.1.1</p> <p>Fixed in Release: 4.7-1.0.0.1</p>
1817636	<p>Description: Fixed the issue of when disabling one port on the Server side, VF-LAG Tx Affinity would not work on the Client side.</p> <p>Keywords: VF-LAG, Tx Affinity</p> <p>Discovered in Release: 4.6-1.0.1.1</p> <p>Fixed in Release: 4.7-1.0.0.1</p>
1843020	<p>Description: Server reboot may result in a system crash.</p> <p>Keywords: reboot, crash</p> <p>Discovered in Release: 4.2-1.2.0.0</p> <p>Fixed in Release: 4.7-1.0.0.1</p>
1811973	<p>Description: VF mirroring offload is now supported.</p> <p>Keywords: ASAP², VF mirroring</p> <p>Discovered in Release: 4.6-1.0.1.1</p> <p>Fixed in Release: 4.7-1.0.0.1</p>
1841634	<p>Description: The number of guaranteed counters per VF is now calculated based on the number of ports mapped to that VF. This allows more VFs to have counters allocated.</p> <p>Keywords: Counters, VF</p> <p>Discovered in Release: 4.4-1.0.0.0</p> <p>Fixed in Release: 4.7-1.0.0.1</p>
1523548	<p>Description: Fixed the issue where RDMA connection persisted even after dropping the network interface.</p> <p>Keywords: Network interface, RDMA</p> <p>Discovered in Release: 4.4-1.0.0.0</p> <p>Fixed in Release: 4.6-1.0.1.1</p>

Internal Reference Number	Issue
1712870	<p>Description: Fixed the issue where small packets with non-zero padding were wrongly reported as "checksum complete" even though the padding was not covered by the csum calculation. These packets now report "checksum unnecessary". In addition, an ethtool private flag has been introduced to control the "checksum complete" feature: <code>ethtool --set-priv-flags eth1 rx_no_csum_complete on/off</code></p> <p>Keywords: csum error, checksum, mlx5_core</p> <p>Discovered in Release: 4.5-1.0.1.0</p> <p>Fixed in Release: 4.6-1.0.1.1</p>
1648597	<p>Description: Fixed the wrong wording in the FW tracer ownership startup message (from "FW Tracer Owner" to "FWTracer: Ownership granted and active").</p> <p>Keywords: FW Tracer</p> <p>Discovered in Release: 4.5-1.0.1.0</p> <p>Fixed in Release: 4.6-1.0.1.1</p>
1581631	<p>Description: Fixed the issue where GID entries referenced to by a certain user application could not be deleted while that user application was running.</p> <p>Keywords: RoCE, GID</p> <p>Discovered in Release: 4.5-1.0.1.0</p> <p>Fixed in Release: 4.6-1.0.1.1</p>
1403313	<p>Description: Fixed the issue of when attempting to allocate an excessive number of VFs per PF in operating systems with kernel versions below v4.15, the allocation failed due to a known issue in the Kernel.</p> <p>Keywords: VF, PF, IOMMU, Kernel, OS</p> <p>Discovered in Release: 4.5-1.0.1.0</p> <p>Fixed in Release: 4.6-1.0.1.1</p>
1368390	<p>Description: Fixed the issue where MLNX_EN could not be installed on RHEL 7.x Alt OSs using YUM repository.</p> <p>Keywords: Installation, YUM, RHEL</p> <p>Discovered in Release: 4.3-3.0.2.1</p> <p>Fixed in Release: 4.6-1.0.1.1</p>
1531817	<p>Description: Fixed an issue of when the number of channels configured was less than the number of CPUs available, part of the CPUs would not be used by Tx queues.</p> <p>Keywords: Performance, Tx, CPU</p> <p>Discovered in Release: 4.4-1.0.1.0</p> <p>Fixed in Release: 4.5-1.0.1.0</p>
1400381	<p>Description: Fixed the issue where on SLES 11 SP3 PPC64 OSs, a memory allocation issue might prevent the interface from loading after reboot, resulting in a call trace in the message log.</p> <p>Keywords: SLES11 SP3</p> <p>Discovered in Release: 4.4-1.0.1.0</p> <p>Fixed in Release: 4.5-1.0.1.0</p>

Internal Reference Number	Issue
1498931	<p>Description: Fixed the issue where establishing TCP connection took too long due to failure of SA PathRecord query callback handler.</p> <p>Keywords: TCP, SA PathRecord</p> <p>Discovered in Release: 4.4-1.0.1.0</p> <p>Fixed in Release: 4.5-1.0.1.0</p>
1514096	<p>Description: Fixed the issue where lack of high order allocations caused driver load failure. All high order allocations are now changed to order-0 allocations.</p> <p>Keywords: mlx5, high order allocation</p> <p>Discovered in Release: 4.4-2.0.0.1</p> <p>Fixed in Release: 4.5-1.0.1.0</p>
1524932	<p>Description: Fixed a backport issue on some OSs, such as RHEL v7.x, where mlx5 driver would support <i>ip link set DEVICE vf NUM rate TXRATE</i> old command, instead of <i>ip link set DEVICE vf NUM max_tx_rate TXRATE min_tx_rate TXRATE</i> new command.</p> <p>Keywords: mlx5 driver</p> <p>Discovered in Release: 4.4-2.0.0.1</p> <p>Fixed in Release: 4.5-1.0.1.0</p>
1498585	<p>Description: Fixed the issue of when performing configuration changes, mlx5e counters values were reset.</p> <p>Keywords: Ethernet counters</p> <p>Discovered in Release: 4.4-2.0.0.1</p> <p>Fixed in Release: 4.5-1.0.1.0</p>
1484603	<p>Description: Fixed the issue of when using <i>ibv_exp_cqe_ts_to_ns</i> verb to convert a packet's hardware timestamp to UTC time in nanoseconds, the result may appear backwards compared to the converted time of a previous packet.</p> <p>Keywords: libibverbs</p> <p>Discovered in Release: 4.4-1.0.1.0</p> <p>Fixed in Release: 4.5-1.0.1.0</p>
1425027	<p>Description: Fixed the issue where attempting to establish a RoCE connection on the default GID or on IPv6 link-local address might have failed when two or more netdevices that belong to HCA ports were slaves under a bonding master. This might also have resulted in the following error message in the kernel log: <i>"__ib_cache_gid_add: unable to add gid fe80:0000:0000:0000:f652:14ff:fe46:7391 error=-28"</i>.</p> <p>Keywords: RoCE, bonding</p> <p>Discovered in Release: 4.4-1.0.1.0</p> <p>Fixed in Release: 4.5-1.0.1.0</p>
1412468	<p>Description: Added support for multi-host connection on mstflint's mstfwreset.</p> <p>Keywords: mstfwreset, mstflint, MFT, multi-host</p> <p>Discovered in Release: 4.3-1.0.1.0</p> <p>Fixed in Release: 4.4-1.0.1.0</p>

Internal Reference Number	Issue
1423319	<p>Description: Removed the following prints on server shutdown: mlx5_core 0005:81:00.1: mlx5_enter_error_state:96:(pid1): start mlx5_core 0005:81:00.1: mlx5_enter_error_state:109:(pid1): end</p> <p>Keywords: mlx5, fast shutdown</p> <p>Discovered in Release: 4.3-1.0.1.0</p> <p>Fixed in Release: 4.4-1.0.1.0</p>
1318251	<p>Description: Fixed the issue of when bringing mlx5 devices up or down, a call trace in <i>nvme_rdma_remove_one</i> or <i>nvmet_rdma_remove_one</i> may occur.</p> <p>Keywords: NVMeoF, mlx5, call trace</p> <p>Discovered in Release: 4.3-1.0.1.0</p> <p>Fixed in Release: 4.4-1.0.1.0</p>
1247458	<p>Description: Added support for VLAN Tag (VST) creation on RedHat v7.4 with new iproute2 packages (iptool).</p> <p>Keywords: SR-IOV, VST, RedHat</p> <p>Discovered in Release: 4.2-1.2.0.0</p> <p>Fixed in Release: 4.3-1.0.1.0</p>
1229554	<p>Description: Enabled RDMA CM to honor incoming requests coming from ports of different devices.</p> <p>Keywords: RDMA CM</p> <p>Discovered in Release: 4.2-1.0.0.0</p> <p>Fixed in Release: 4.3-1.0.1.0</p>
1262257	<p>Description: Fixed an issue where sending Work Requests (WRs) with multiple entries where the first entry is less than 18 bytes used to fail.</p> <p>Keywords: ConnectX-5, libibverbs, Raw QP</p> <p>Discovered in Release: 4.2-1.2.0.0</p> <p>Fixed in Release: 4.3-1.0.1.0</p>
1249358/1261023	<p>Description: Fixed the issue of when the interface was down, ethtool counters ceased to increase. As a result, RoCE traffic counters were not always counted.</p> <p>Keywords: Ethtool counters, mlx5</p> <p>Discovered in Release: 4.2-1.2.0.0</p> <p>Fixed in Release: 4.3-1.0.1.0</p>
1244509	<p>Description: Fixed compilation errors of MLNX_EN over kernel when CONFIG_PTP_1588_CLOCK parameter was not set.</p> <p>Keywords: PTP, mlx5e</p> <p>Discovered in Release: 4.2-1.2.0.0</p> <p>Fixed in Release: 4.3-1.0.1.0</p>
1266802	<p>Description: Fixed an issue where the system used to hang when trying to allocate multiple device memory buffers from different processes simultaneously.</p> <p>Keywords: Device memory programming</p>

Internal Reference Number	Issue
	<p>Discovered in Release: 4.2-1.0.0.0</p> <p>Fixed in Release: 4.3-1.0.1.0</p>
1078887	<p>Description: Fixed an issue where <code>post_list</code> and <code>CQ_mod</code> features in <code>perftest</code> did not function when running the <code>--run_infinitely</code> flag.</p> <p>Keywords: <code>perftest</code>, <code>--run_infinitely</code></p> <p>Discovered in Release: 4.2-1.0.1.0</p> <p>Fixed in Release: 4.2-1.2.0.0</p>
1186260	<p>Description: Fixed the issue where CNP counters exposed under <code>/sys/class/infiniband/mlx5_bond_0/ports/1/hw_counters/</code> did not aggregate both physical functions when working in RoCE LAG mode.</p> <p>Keywords: RoCE, LAG, ECN, Congestion Counters</p> <p>Discovered in Release: 4.2-1.0.1.0</p> <p>Fixed in Release: 4.2-1.2.0.0</p>
1192374	<p>Description: Fixed wrong calculation of <code>max_device_ctx</code> capability in ConnectX-4, and ConnectX-5 HCAs.</p> <p>Keywords: <code>ibv_exp_query_device</code>, <code>max_device_ctx</code> <code>mlx5</code></p> <p>Discovered in Release: 4.2-1.0.1.0</p> <p>Fixed in Release: 4.2-1.2.0.0</p>
1084791	<p>Description: Fixed the issue where occasionally, after reboot, <code>rpm</code> commands used to fail and create a core file, with messages such as “Bus error (core dumped)”, causing the <code>openibd</code> service to fail to start.</p> <p>Keywords: <code>rpm</code>, <code>openibd</code></p> <p>Discovered in Release: 3.4-2.0.0.0</p> <p>Fixed in Release: 4.2-1.0.1.0</p>
960642/960653	<p>Description: Added support for <code>min_tx_rate</code> and <code>max_tx_rate</code> limit per virtual function ConnectX-5 and ConnectX-5 Ex adapter cards.</p> <p>Keywords: SR-IOV, <code>mlx5</code></p> <p>Discovered in Release: 4.0-1.0.1.0</p> <p>Fixed in Release: 4.2-1.0.1.0</p>
866072/869183	<p>Description: Fixed the issue where RoCE v2 multicast traffic using RDMA-CM with IPv4 address was not received.</p> <p>Keywords: RoCE</p> <p>Discovered in Release: 3.4-1.0.0.0</p> <p>Fixed in Release: 4.2-1.0.1.0</p>
1163835	<p>Description: Fixed an issue where <code>ethtool -P</code> output was <code>00:00:00:00:00:00</code> when using old kernels.</p> <p>Keywords: <code>ethtool</code>, Permanent MAC address, <code>mlx5</code></p> <p>Discovered in Release: 4.0-2.0.0.1</p> <p>Fixed in Release: 4.2-1.0.1.0</p>

Internal Reference Number	Issue
1067158	<p>Description: Replaced a few “GPL only” legacy libibverbs functions with upstream implementation that conforms with libibverbs GPL/BSD dual license model.</p> <p>Keywords: libibverbs, license</p> <p>Discovered in Release: 4.1-1.0.2.0</p> <p>Fixed in Release: 4.2-1.0.1.0</p>
1119377	<p>Description: Fixed an issue where ACCESS_REG command failure used to appear upon RoCE Multihost driver restart in dmesg. Such an error message looked as follows: <i>mlx5_core 0000:01:00.0: mlx5_cmd_check:705:(pid 20037): ACCESS_REG(0x805) op_mod(0x0) failed, status bad parameter(0x3), syndrome (0x15c356)</i></p> <p>Keywords: RoCE, multihost, mlx5</p> <p>Discovered in Release: 4.1-1.0.2.0</p> <p>Fixed in Release: 4.2-1.0.1.0</p>
1122937	<p>Description: Fixed an issue where concurrent client requests got corrupted when working in persistent server mode due to a race condition on the server side.</p> <p>Keywords: librdmacm, rping</p> <p>Discovered in Release: 4.1-1.0.2.0</p> <p>Fixed in Release: 4.2-1.0.1.0</p>
1102158	<p>Description: Fixed an issue where client side did not exit gracefully in RTT mode when the server side was not reachable.</p> <p>Keywords: librdmacm, rping</p> <p>Discovered in Release: 4.1-1.0.2.0</p> <p>Fixed in Release: 4.2-1.0.1.0</p>
1038933	<p>Description: Fixed a backport issue where IPv6 procedures were called while they were not supported in the underlying kernel.</p> <p>Keywords: iw_cm</p> <p>Discovered in Release: 4.0-2.0.0.1</p> <p>Fixed in Release: 4.1-1.0.2.0</p>
1064722	<p>Description: Added log debug prints when changing HW configuration via DCB. To enable log debug prints, run: <i>ethtool -s <devname> msglvl hw on/off</i></p> <p>Keywords: DCB, msglvl</p> <p>Discovered in Release: 4.0-2.0.0.1</p> <p>Fixed in Release: 4.1-1.0.2.0</p>
1047617	<p>Description: Fixed the issue where a race condition in the RoCE GID cache used to cause for the loss of IP-based GIDs.</p> <p>Keywords: RoCE, GID</p> <p>Discovered in Release: 4.0-2.0.0.1</p> <p>Fixed in Release: 4.1-1.0.2.0</p>
1006768	<p>Description: Fixed the issue where an rdma_cm connection between a client and a server that were on the same host was not possible when working over VLAN interfaces.</p>

Internal Reference Number	Issue
	<p>Keywords: RDMACM</p> <p>Discovered in Release: 4.0-2.0.0.1</p> <p>Fixed in Release: 4.1-1.0.2.0</p>
801807	<p>Description: Fixed an issue where RDMACM connection used to fail upon high connection rate accompanied with the error message: <i>RDMA_CM_EVENT_UNREACHABLE</i> .</p> <p>Keywords: RDMACM</p> <p>Discovered in Release: 3.0-2.0.1</p> <p>Fixed in Release: 4.1-1.0.2.0</p>
869768	<p>Description: Fixed the issue where SR-IOV was not supported in systems with a page size greater than 16KB.</p> <p>Keywords: SR-IOV, mlx5, PPC</p> <p>Discovered in Release: 4.0-2.0.0.1</p> <p>Fixed in Release: 4.1-1.0.2.0</p>
919545	<p>Description: Fixed the issue of when the Kernel becomes out of memory upon driver start, it could crash on SLES 12 SP2.</p> <p>Keywords: mlx_5 Eth Driver</p> <p>Discovered in Release: 3.4-2.0.0.0</p> <p>Fixed in Release: 4.0-2.0.0.1</p>
869209	<p>Description: Fixed an issue that caused TCP packets to be received in an out of order manner when Large Receive Offload (LRO) is on.</p> <p>Keywords: mlx5_en</p> <p>Discovered in Release: 3.3-1.0.0.0</p> <p>Fixed in Release: 4.0-2.0.0.1</p>
890285	<p>Description: Fixed the issue where memory allocation for CQ buffers used to fail when increasing the RX ring size.</p> <p>Keywords: mlx5_core</p> <p>Discovered in Release: 3.4-1.0.0.0</p> <p>Fixed in Release: 4.0-1.0.1.0</p>
867094	<p>Description: Fixed the issue where MLNX_EN used to fail to load on 4K page Arm architecture.</p> <p>Keywords: Arm</p> <p>Discovered in Release: 3.4-1.0.0.0</p> <p>Fixed in Release: 4.0-1.0.1.0</p>

3.2 User Manual Revision History

Release	Date	Description
5.7	August 2022	<ul style="list-style-type: none"> Added Out of Order (OOO) under RoCE section
5.3	April 15, 2021	<ul style="list-style-type: none"> Added PTP Cyc2time Hardware Translation Offload section Updated Persistent Naming section
5.2	January 12, 2021	<ul style="list-style-type: none"> Added Offloaded Traffic Sniffer section. Added Tx Port Time-Stamping section. Added VLAN Push/Pop section. Added sFLOW section. Added E2E Cache section. Added Geneve Encapsulation/Decapsulation section. Added Parallel Offloads section. Updated SR-IOV VF LAG section. Removed Installing MLNX_EN on Innova™ IPsec Adapter Cards section. Removed Updating Firmware and FPGA Image on Innova IPsec Cards section.
5.1	August 16, 2020	<p>Updated the content of the entire document following the removal of support for ConnectX-3, ConnectX-3 Pro and Connect-IB adapter cards, as well as the deprecation of RDMA experimental verbs library (mlx_lib).</p> <p>Added Interrupt Request (IRQ) Naming section.</p> <p>Added Kernel Transport Layer Security (kTLS) Offloads section.</p>
5.0	March 15, 2020	<ul style="list-style-type: none"> Added IPsec Crypto Offload section. Updated Installing MLNX_EN v4.5-1.0.1.0 section.
4.7	December 29, 2019	<ul style="list-style-type: none"> Added section .Mediated Devices v4.7-3.2.9.0. Added "num_of_groups" entry to table mlx5_core Module Parameters. Added Performance Tuning Based on Traffic Patterns section.
4.5	December 19, 2018	<ul style="list-style-type: none"> Reorganized Chapter 2, "Installation": Consolidated the separate installation procedures under Installing MLNX_EN and Additional Installation Procedures

4 Legal Notices and 3rd Party Licenses

The following are the drivers' software, tools and HCA firmware legal notices and 3rd party licenses.

Product	Version	Legal Notices and 3rd Party Licenses
MLNX_OFED	5.8-3.0.7.0	<ul style="list-style-type: none">• License• 3rd Part Notice
Firmware	xx.35.3006	<ul style="list-style-type: none">• HCA Firmware EULA• License• 3rd Party Notice
MFT	4.22.1-307	<ul style="list-style-type: none">• License• 3rd Party Notice
Clusterkit	2.16	<ul style="list-style-type: none">• License• 3rd Party Notice
DPCP	1.1.40	<ul style="list-style-type: none">• License• 3rd Party Notice
VMA	9.8.31	<ul style="list-style-type: none">• 3rd Party Unify Notice• 3rd Party Notice
XLIO	3.10	<ul style="list-style-type: none">• License• 3rd Party Unify Notice
HCOLL	2.16	<ul style="list-style-type: none">• License• 3rd Party Notice
SHARP	2.16	<ul style="list-style-type: none">• License• 3rd Party Notice
ibutils2	2.14	<ul style="list-style-type: none">• License• 3rd Party Notice
OpenSM	5.16	<ul style="list-style-type: none">• 3rd Party Unify Notice• 3rd Party Notice
mpitests	2.16	<ul style="list-style-type: none">• License• 3rd Party Notice

Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. Neither NVIDIA Corporation nor any of its direct or indirect subsidiaries and affiliates (collectively: "NVIDIA") make any representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice. Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer ("Terms of Sale"). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

NVIDIA products are not designed, authorized, or warranted to be suitable for use in medical, military, aircraft, space, or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death, or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer's own risk.

NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer's sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer's product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this document. NVIDIA accepts no liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this document or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA.

Reproduction of information in this document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

THIS DOCUMENT AND ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA's aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms of Sale for the product.

Trademarks

NVIDIA, the NVIDIA logo, and Mellanox are trademarks and/or registered trademarks of NVIDIA Corporation and/or



Mellanox Technologies Ltd. in the U.S. and in other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

© 2024 NVIDIA Corporation & affiliates. All Rights Reserved.

