
# OVS Offload using ASAP$^2$ Direct on ConnectX-4 / ConnectX-4 Lx Release Notes

Rev 2.0


Mellanox Technologies
350 Oakmead Parkway Suite 100
Sunnyvale, CA 94085
U.S.A.
www.mellanox.com
Tel: (408) 970-3400
Fax: (408) 970-3403

# Table of Contents

# List of Tables

# Release Update History

***Table 1 - Release Update History***

| Release | Date | Description |
|---|---|---|
| 2.0 | July 2016 | Initial release of this version |

# 1 Overview

These are the release notes of Open vSwitch (OVS) offload using Mellanox "Accelerated Switching And Packet Processing" (ASAP$^2$) Direct technology for Linux and ConnectX-4 / ConnectX-4 Lx cards.

This release was optimized for up to 10000 flows. Future releases will further optimize current performance and address larger flow count.

## 1.1 Package Contents

OVS Offload using ASAP$^2$ Direct for Linux and ConnectX-4/ConnectX-4 Lx cards software contains the following components:

- Firmware (for the supported versions, please refer to Section 1.2.3, "Firmware Versions", on page 4)
- Host kernel DEB/RPM v4.2.3_ovs_offload_r24 (based on upstream kernel 4.2.x)
- Mellanox OFED package for the VM, v3.2-2.0.3.0.7
- MLNX_DPDK_2.2.0-MR

## 1.2 Supported Platforms and Operating Systems

### 1.2.1 Supported Hypervisors

The following are the supported Hypervisors distributions (KVM/x86_64) over which the OVS kernel should be installed:

***Table 2 - Supported Hypervisors (KVM / x86_64)***

| Operating System | Platform |
|---|---|
| RHEL7.2 | x86_64 |
| Ubuntu 15.04 | x86_64 |
| Ubuntu 15.10 | x86_64 |

### 1.2.2 Supported Guest OSs

The following are the supported Guest distributions (x86_64) over which MLNX OFED and DPDK should be installed:

***Table 3 - Supported Guest (x86_64)***

| Operating System | Platform |
|---|---|
| RHEL7.2 | x86_64 |
| Ubuntu 15.04 | x86_64 |
| Ubuntu 15.10 | x86_64 |

### 1.2.3 Firmware Versions

OVS Offload using ASAP$^2$ Direct on ConnectX-4/ConnectX-4 Lx Rev 2.0 requires the following firmware for Mellanox NICs:

***Table 4 - Firmware Versions***

| NICs | Recommended Firmware Rev. | Additional Firmware Rev. Supported |
|---|---|---|
| ConnectX®-4 | 12.16.2204 (at beta level) | N/A |
| ConnectX®-4 Lx | 14.16.2204 (at beta level) | N/A |

# 2 Changes and New Features in Rev 2.0

***Table 5 - Changes and New Feature in Rev 2.0***

| Feature/Change | Description |
|---|---|
| **Offloading Scalability Enhancements** | Added support for offloading large amounts of flows (optimized for up to ten thousand flows) by optimizing firmware and driver. This includes lockless and bulking approaches for efficient gathering of HW flow statistics under a large scale of offloaded flows in the driver, and optimized tables and rules setup in the firmware. |

# 3 Known Issues

The following table provides a list of known issues in regards to this release.

***Table 6 - Known Issues***

| Internal Ref. | Issue |
|---|---|
| 768065 | **Description:** On some VF drivers, the VF MTU cannot be reduced below the current PF MTU. |
| | **WA**: Return the Physical Function (PF) MTU to 1500 bytes after reducing the VF MTU. |
| | **Keywords:** MTU |
| - | **Description:** libmlx5 (e.g used by DPDK) running over VFs is not aware to the required minimal inline size as configured by the PF, unless directed by the VM user.<br><br>**Note:** This value needs to be set according to the "flow_offload_min_inline" parameter configured on the Hypervisor as the HW will enforce it.<br>This means:<br>• Kernel: The driver in MLNX_OFED v3.2-2.0.3.0.3 package will perform this automatically<br>• Userspace: The SW needs to be aware of it and copy headers inline accordingly. This parameter will perform it in case of DPDK. |
| | **WA**:<br>If non-default (L3 or L4) matching on outgoing VF traffic is required, export the MLX5_WQE_MIN_INLINE_SIZE=N environment variable to the processes running libmlx5. Where N is the number of header bytes needed for the required matching:<br>• for L3 matching use 34 (14+20)<br>• for L4 UDP port matching use 42 (14+20+8)<br>• for L4 TCP port matching use 74 (14+20+40) if TCP time-stamping is enabled and<br>• 66 (14+20+32) if TCP TS is disabled |
| | **Keywords:** libmlx5, minimal inline size |
| - | **Description:** Learning a few thousand new flows per second may not work properly. |
| | **WA**: Avoid getting OVS to learn over 2000 new flows per second |
| | **Keywords:** Offload flows |

# 4 Bug Fixes History

Table 7 lists the bugs fixed in this release.

***Table 7 - Fixed Bugs List***

| Internal Ref. | Issue |
|---|---|
| 758724 | **Description:** Fixed the issue where MLNX_OFED v3.2 kernel driver failed to start Virtual Function (VF) netdevs when the host ran in OVS offloads mode. |
| | **Keywords:** VF netdevs, MLNX_OFED, OVS offloads mode |
| | **Discovered in Release:** 1.0 |
| | **Fixed in Release:** 2.0 |
| 766780 | **Description:** Fixed the issue where learning new flows under very heavy packet rate might have not worked properly. |
| | **Keywords:** Heavy packet rate, Offload flows |
| | **Discovered in Release:** 1.0 |
| | **Fixed in Release:** 2.0 |

# 5 Change Log History

***Table 8 - History of Changes and New Features***

| Rev. | Feature/ Change | Description |
|---|---|---|
| **1.0** | **OVS Offload** | Mellanox Accelerated Switching And Packet Processing (ASAP$^2$) Direct technology allows to offload OVS by handling OVS data-plain in Mellanox ConnectX-4/ ConnectX-4 Lx NIC hardware (Mellanox Embedded Switch or eSwitch) while maintaining OVS control-plain unmodified. The current actions supported by ASAP$^2$ Direct include packet parsing and matching, forward, drop along with VLAN push/pop or VXLAN encap/decap and HW based packet/byte flow statistics. |
| | **VXLAN encap/decap** | Virtual Extensible LAN (VXLAN) is a network virtualization technology that improves scalability problems associated with large cloud computing deployments. It tunnels Ethernet frames within Ethernet + IP + UDP frames. Mellanox implements VXLAN encapsulation and decapsulation in hardware. |

# 6 Offload Limitations

The following is a list of offload limitations in regards to this release.

- Changing SR-IOV e-switch offloads mode (on --> off and off --> on) requires all VFs to be unbinded
- VXLAN encap/decap offloading of OVS actions is supported only in ConnectX-4 Lx
- UDP destination port for offloading VXLAN must be 4789
- When offloading VXLAN, the Guest MTU must be reduced to account for the encap headers (1500 --> 1450), as IP fragmentation for encap offloads is not supported. If this is not applied, VF packets are silently dropped at the port
- RoCE for VFs is not supported when VXLAN encap/decap offloads are enabled
- FIB updates are not supported for VXLAN offloads. If a flow is offloaded and then the IP of the PF is changed, it will not be un-offloaded, and vice versa
- Only a single VLAN per VM is supported for offloading VLAN push/pop actions
- Offloaded flows must match on the destination MAC
- Using OVS in a non-learning manner (e.g by deleting the Open-Flow normal rule) causes packets that do not match any of the rules to be dropped, and not be offloaded because the derived data-path rule does not match on dmac
- Offloading matches on L3 or L4 (TCP or UDP) headers for outgoing flows (the source port is a VF, not wire), requires administrative directive to the PF driver to set "min inline size" for the VFs
- Flows containing the following matching attributes are not offloaded: ARPs, IPv6, TCP flags, IP fragment type