NVIDIA BlueField DPU Modes of Operation

User Guide

MLNX-15-060472 _v1.3 | June 2022

Table of Contents

Chapter 1. Introduction

The NVIDIA® BlueField® DPU has several modes of operation:

- Embedded function (ECPF)or embedded function (ECPF) ownership where the embedded Arm system controls the NIC resources and data path (**default**)
- Restricted mode which is an extension of the ECPF ownership with additional restrictions on the host side
- NIC Mode where the DPU behaves exactly like an adapter card from the perspective of the external host
- Separated host mode (symmetric model)

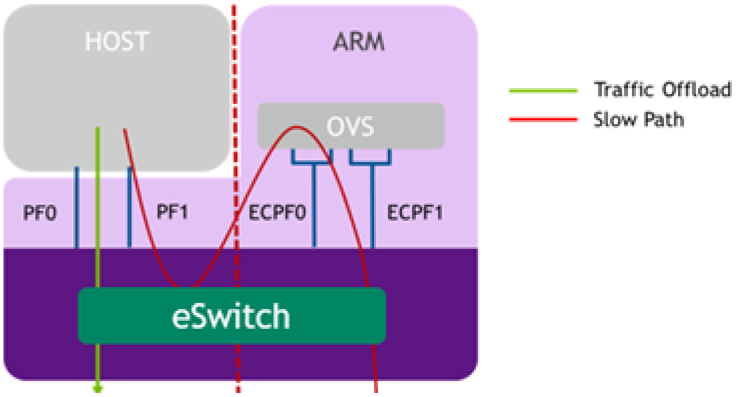
Chapter 2. DPU Mode

This mode, also known as ECPF or DPU mode, is the default mode for BlueField DPU.

In DPU mode, the NIC resources and functionality are owned and controlled by the embedded Arm subsystem. All network communication to the host flows through a virtual switch control plane hosted on the Arm cores, and only then proceeds to the x86 host. While working in this mode, the DPU is the trusted function managed by the data center and host administrator—to load network drivers, reset an interface, bring an interface up and down, update the firmware, and change the mode of operation on the DPU device.

A network function is still exposed to the host, but it has limited privileges. In particular:

1. The driver on the host side can only be loaded after the driver on the embedded side has loaded and completed NIC configuration.
2. All ICM (Interface Configuration Memory) is allocated by the ECPF and resides in the embedded host memory.
3. The ECPF controls and configures the NIC embedded switch which means that traffic to and from the host interface always lands on the Arm side.



When the server and DPU are initiated, the networking to the host is blocked until the virtual switch on the DPU is loaded. Once it is loaded, traffic to the host is allowed by default.

There are two ways to pass traffic to the host interface: Either using representors to forward traffic to the host (every packet to/from the host would be handled also by the network interface on the embedded Arm side), or push rules to the embedded switch which allows and offloads this traffic.

2.1. Configuring DPU Mode

To enable this mode:

1. Start MST driver set service:

```
$ mst start
```

2. Identify the MST device:

```
$ mst status -v
```

Output example:

```
...
DEVICE_TYPE             MST                          PCI         RDMA
 NET                       NUMA
BlueField(rev:0)        /dev/mst/<device>.1        37:00.1     mlx5_1
 net-ens1f1                0

BlueField(rev:0)        /dev/mst/<device>          37:00.0     mlx5_0
 net-ens1f0                0
```

3. Run the following commands on the Arm:

```
$ mlxconfig -d /dev/mst/<device> s INTERNAL_CPU_MODEL=1
```

4. Power cycle the server.

> **Note:** If OVS bridges `ovsbr1` and `ovsbr2` are not created (you may verify that using command `ovs-vsctl show`), make sure field `CREATE_OVS_BRIDGES` is set to `"yes"` in `/etc/mellanox/mlnx-ovs.conf`.

Chapter 3. Restricted DPU Host Mode

Restricted mode is a specialization of Embedded mode and implements an additional layer of security where the host system administrator is prevented from accessing the DPU from the host. Once Restricted mode is enabled, the data center administrator should control the DPU entirely though the Arm cores and/or BMC connection instead of through the host.

For security and isolation purposes, it is possible to restrict the host from performing operations that can compromise the DPU. The following operations can be restricted individually when changing the DPU host to Restricted mode:

- Port ownership – the host cannot assign itself as port owner
- Hardware counters – the host does not have access to hardware counters
- Tracer functionality is blocked
- RShim interface is blocked
- FW flash is restricted

3.1. Enabling Host Restriction

To enable host restriction:

1. Start the MST service.

   ```
   $ mst start
   ```

2. Set restricted mode. From the Arm side, run:

   ```
   $ mlxprivhost -d /dev/mst/<device> r --disable_rshim --disable_tracer --
   disable_counter_rd --disable_port_owner
   ```

   Note: If RShim is disabled, power cycle is required.

   Note: Power cycle is required if any `--disable_*` flags are used.

3.2. Disabling Host Restriction

To disable host restriction set the mode to privileged mode:

```
$ mlxprivhost -d /dev/mst/<device> p
```

The configuration takes effect immediately.

Note: If you are reverting from `rshim-disabled` mode, system power cycle is required.

Note: Power cycle is required when reverting to privileged mode if host restriction has been applied using any `--disable_*` flags.

3.3. Hiding Network Ports from Host

1. Make sure the DPU is operating in embedded mode. Run:

   ```
   # mlxconfig -d /dev/mst/<device> q | grep -i cpu
   INTERNAL_CPU_MODEL EMBEDDED_CPU(1)
   ```

   If the output is `INTERNAL_CPU_MODEL SEPERATED_HOST(0)`, it means that your device is operating in separated mode. See section Configuring ECPF Mode from Separated Host Mode to change this configuration.

2. To hide network ports from host, run:

   ```
   mlxconfig -d /dev/mst/<device> s NUM_OF_PF=0
   ```

   Note: Power cycle is required for this configuration to take effect.

3. To re-expose network ports to host, run:

   ```
   mlxconfig -d /dev/mst/<device> s NUM_OF_PF=2
   ```

   Note: Power cycle is required for this configuration to take effect.

Chapter 4. NIC Mode

Note: Prior to configuring NIC Mode, refer to known issue #3048250 in the NVIDIA DOCA Release Notes.

In this mode, the DPU behaves exactly like an adapter card from the perspective of the external host. The ECPFs on the Arm side are not functional in this mode but the user is still able to access the Arm system and update `mlxconfig` options.

To enable DPU NIC mode, run the following from the x86 host side:

```
$ mst start
$ mlxconfig -d /dev/mst/<device> s INTERNAL_CPU_MODEL=1 \
INTERNAL_CPU_PAGE_SUPPLIER=1 \
INTERNAL_CPU_ESWITCH_MANAGER=1 \
INTERNAL_CPU_IB_VPORT0=1 \
INTERNAL_CPU_OFFLOAD_ENGINE=1
$ mlxfwreset -d /dev/mst/<device> r
Minimal reset level for device, /dev/mst/mt41686_pciconf0:

3: Driver restart and PCI reset
Continue with reset?[y/N] y
-I- Sending Reset Command To Fw             -Done
-I- Stopping Driver                         -Done
-I- Resetting PCI                           -Done
-I- Starting Driver                         -Done
-I- Restarting MST                          -Done
-I- FW was loaded successfully.
```

Note: To restrict RShim PF (optional), make sure to configure `INTERNAL_CPU_RSHIM=1` as part of the `mlxconfig` command.

Note: Multi-host is not supported when the DPU is operating in NIC mode.

Note: To obtain firmware BINs for NVIDIA® BlueField®-2 devices, refer to the BlueField-2 firmware download page.

> **Note:** If RShim is disabled, then power cycle is mandatory.

To change from back from NIC mode to DPU (ECPF) mode:

1. Install and start the RShim driver on the x86 host.

2. Disable NIC mode. Run:

```
$ mst start
$ mlxconfig -d /dev/mst/<device> s INTERNAL_CPU_MODEL=1 \
INTERNAL_CPU_PAGE_SUPPLIER=0 \
INTERNAL_CPU_ESWITCH_MANAGER=0 \
INTERNAL_CPU_IB_VPORT0=0 \
INTERNAL_CPU_OFFLOAD_ENGINE=0
$ mlxfwreset -d /dev/mst/<device> r
```

Note: If `INTERNAL_CPU_RSHIM=1`, then make sure to configure `INTERNAL_CPU_RSHIM=0` as part of the `mlxconfig` command.

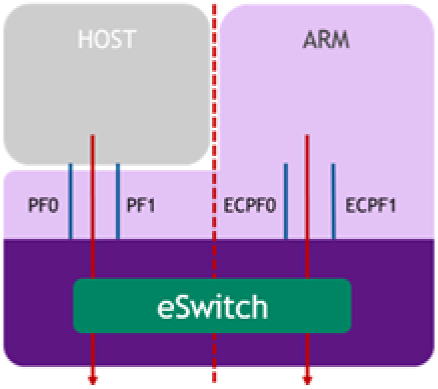
Note: If RShim is enabled, then power cycle is mandatory.

Chapter 5. Separated Host Mode

In Separated mode, a network function is assigned to both the Arm cores and the x86 host cores. The ports/functions are symmetric in the sense that traffic is sent to both physical functions simultaneously. Each one of those functions has its own MAC address, which allows one to communicate with the other, and can send and receive Ethernet and RDMA over Converged Ethernet (RoCE) traffic. There is an equal bandwidth share between the two functions.

There is no dependency between the two functions. They can operate simultaneously or separately. The host can communicate with the embedded function as two separate hosts, each with its own MAC and IP addresses (configured as a standard interface).

In Separated mode, the host administrator is a trusted actor who can perform all configuration and management actions related to either network function.



This mode enables the same operational model of a SmartNIC (that does not have a separated control plane). In this case, the Arm control plane can be used for different functions but does not have any control on the host steering functions.

The limitations of this mode are as follows:

- Switchdev (virtual switch offload) mode is not supported on either of the functions
- SR-IOV is only supported on the host side

5.1. Configuring Separated Host Mode from DPU Mode

On the server host, follow these steps:

1. Enable separated host mode. Run:

```
$ mst start
$ mlxconfig -d /dev/mst/<device> s INTERNAL_CPU_MODEL=0
```

2. Power cycle.
3. Verify configuration. Run:

```
$ mst start
$ mlxconfig -d /dev/mst/<device> q | grep -i model
```

4. Remove OVS bridges configuration from the Arm-side. Run:

```
$ ovs-vsctl list-br | xargs -r -l
```

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. NVIDIA Corporation nor any of its direct or indirect subsidiaries and affiliates (collectively: "NVIDIA") make no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assume 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

Copyright

NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051
http://www.nvidia.com