NVIDIA DOCA NAT

Application Guide

MLNX-15-060555 _v2.0.2 | May 2023

Table of Contents

Chapter 1. Introduction

A network address translation (NAT) application leverages the DPU's hardware capability to switch packets with local IP addresses to global ones and vise versa.

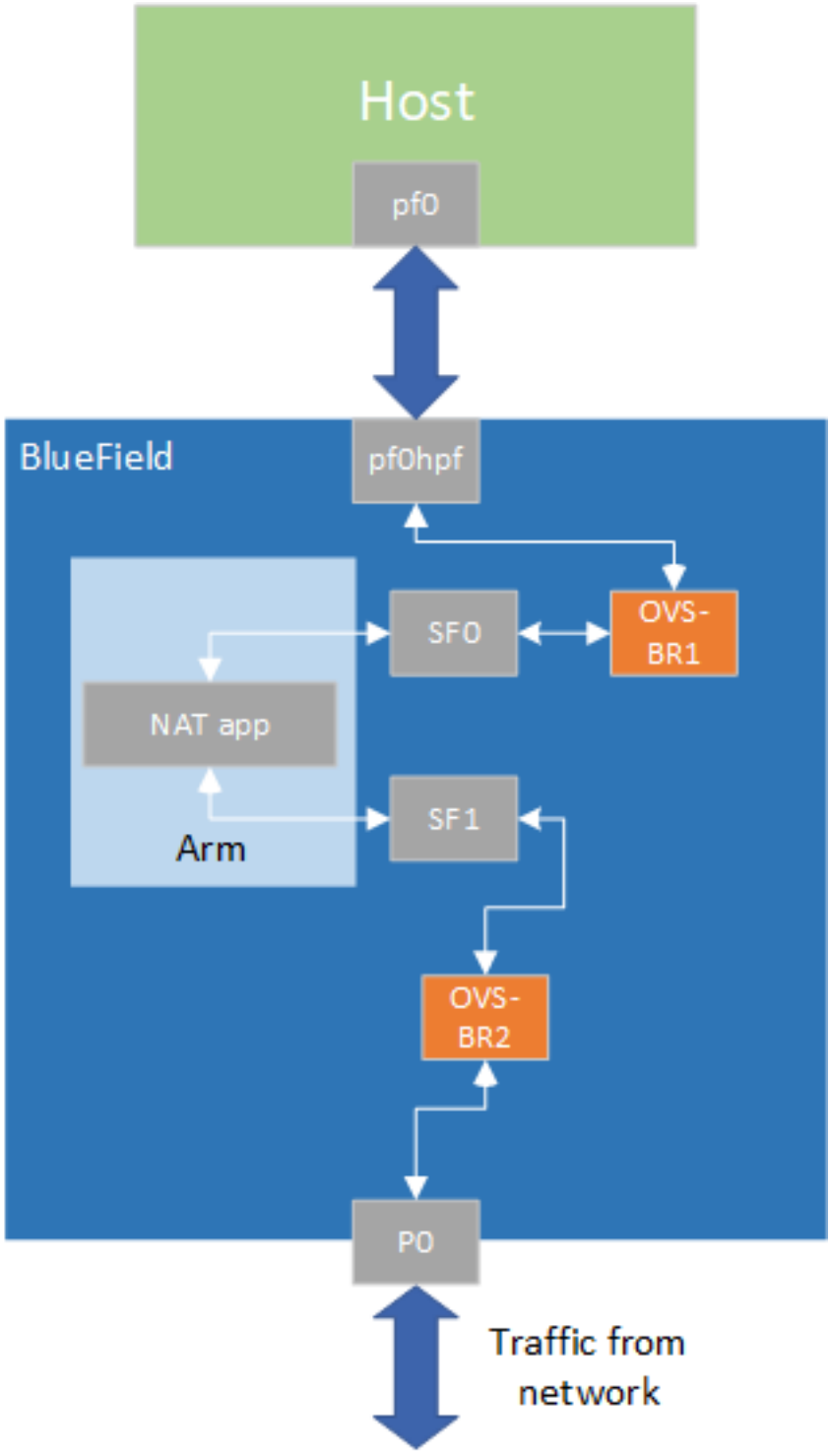
The NAT application is based on DOCA Flow, used for the programming of the DPU's hardware.

NAT can operate in three modes:

- Static mode – application gets pairs of local IP address and global IP address from the user using a JSON file
- Dynamic mode – user provides pool of global IP addresses that can be used. The application should pick 1 address from the pool for new local area network (LAN) IP address and use it. Upon session close, address are returned to the pool.
- PAT mode (DNS offload) – the user provides 1 global address to use. In addition, the user provides mapping between the local port address to the global port. For each packet, the local address is replaced with the global one and ports are replaced according to mapping table.

Chapter 2. System Design

The NAT application is design to run on the DPU. The DPU intercepts ingress traffic from both wire and host, switches the relevant IP address and port according to data configured by the user, and forwards it to the egress port.


Host
pf0
BlueField
pf0hpf
SF0
OVS-
BR1
NAT app
SF1
Arm
OVS-
BR2
P0
Traffic from
network

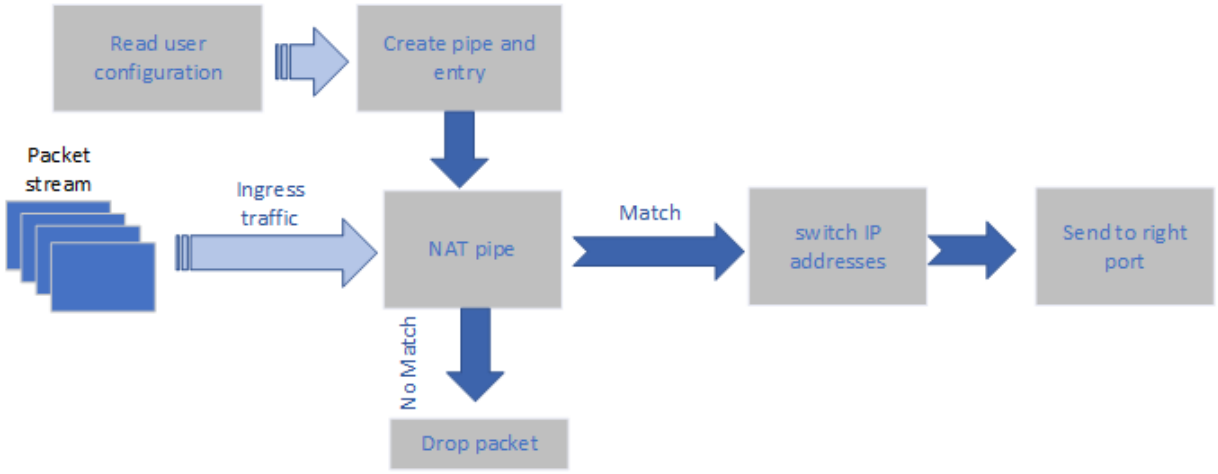
Chapter 3. Application Architecture

NAT runs on the DPU to classify packets.

The app should be configured using a JSON file which includes the operation mode.

3.1. Static Mode

For static mode, the JSON file should include pairs of local and global IP addresses. No change for ports in this mode.

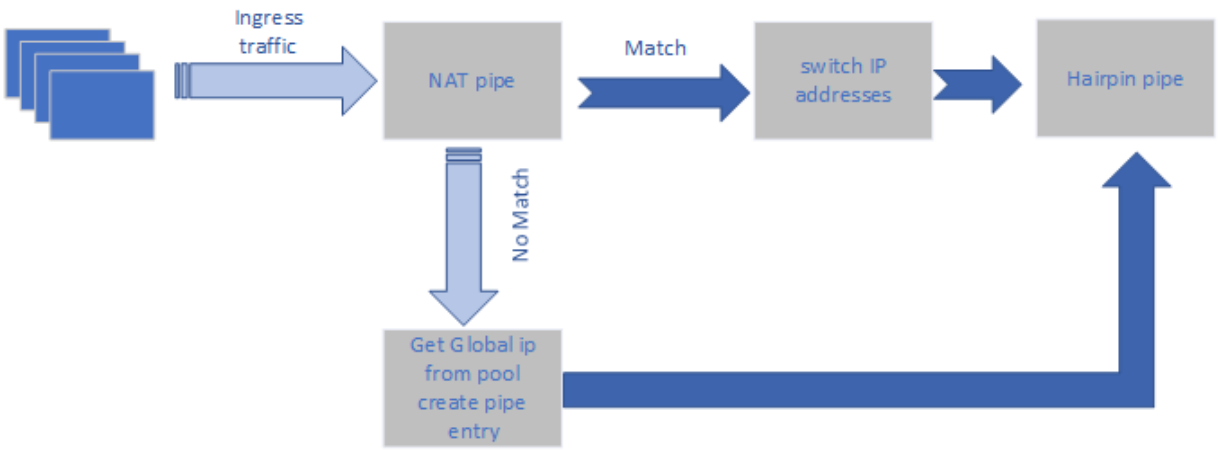


3.2. Dynamic Mode

The user must provide a pool of global IP addresses to use. The application allocates a global address to every miss in the pipe (new local address).

If no more global addresses are available in the pool, the user gets an error message and the packet is sent as is.

The application performs a callback to remove the matching of global and local IPs and returns the address to the pool.

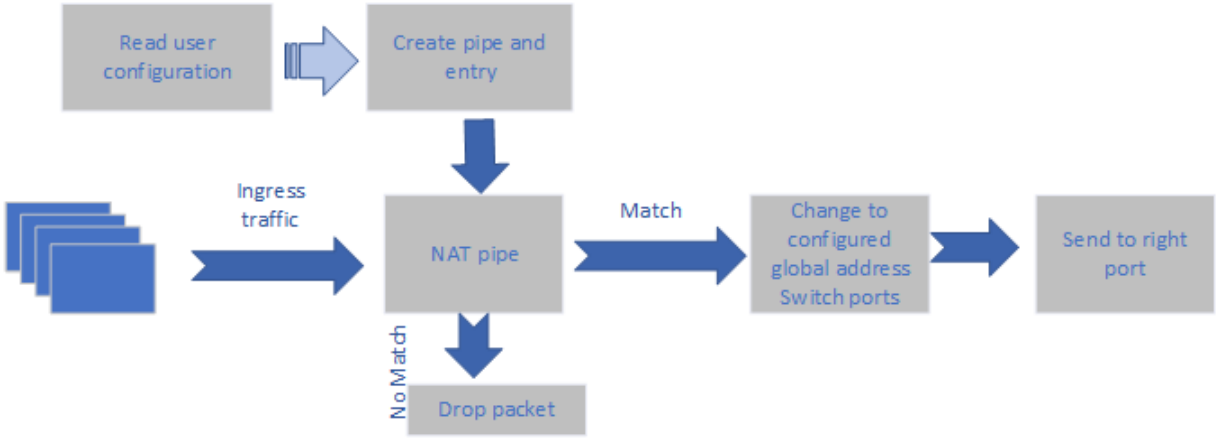


3.3. PAT (NAT Offload) Mode

The user provides a global address to replace all local addresses in the user LAN.

The user provides a matching of local IP and port to global port.

The application changes the local IP of every match to the global IP provided by the user and updates the port number according to user configuration.



Chapter 4. DOCA Libraries

This application leverages the following DOCA libraries:

- DOCA Flow Library

Chapter 5. Configuration Flow

1. Parse application argument.
   a). Initialize arg parser resources and register DOCA general parameters.
   ```
   doca_argp_init();
   ```
   b). Register application parameters.
   ```
   register_nat_params()
   ```
   c). Parse the arguments.
   ```
   doca_argp_start();
   ```
   i. Parse DPDK flags and invoke handler for calling the `rte_eal_init()` function.
   ii. Parse app parameters.
2. DPDK initialization.
   ```
   dpdk_init();
   ```
   Calls `rte_eal_init()` to initialize EAL resources with the provided EAL flags.
3. DPDK port initialization and start.
   ```
   dpdk_queues_and_ports_init();
   ```
   a). Initialize DPDK ports, including mempool allocation.
   b). Initialize hairpin queues if needed.
   c). Bind hairpin queues of each port to its peer port.
4. NAT initialization.
   ```
   nat_init();
   ```
   a). DOCA Flow and DOCA Flow port initialization.
5. Init user configuration rules into app structure.
   ```
   parsing_nat_rules();
   ```
6. Init pipes and entry according to rules.
   ```
   nat_pipes_init
   ```
7. Wait for signal to end application.
8. NAT Destroy.
   ```
   nat_destroy();
   ```
9. DPDK ports and queues destruction.
   ```
   dpdk_queues_and_ports_fini();
   ```
10. DPDK finish.
    ```
    dpdk_fini();
    ```
    a). Calls `rte_eal_destroy()` to destroy initialized EAL resources.

11. Arg parser destroy.

```
doca_argp_destroy();
```

Chapter 6. Running Application

1. Refer to the following documents:
   - NVIDIA DOCA Installation Guide for Linux for details on how to install BlueField-related software.
   - NVIDIA DOCA Troubleshooting Guide for any issue you may encounter with the installation, compilation, or execution of DOCA applications.
   - NVIDIA DOCA Applications Overview for additional compilation instructions and development tips regarding the DOCA applications.
2. The NAT binary is located under `/opt/mellanox/doca/applications/nat/bin/doca_nat`. To build all the applications together, run:

```
cd /opt/mellanox/doca/applications/
meson build
ninja -C build
```

3. To build only the NAT application:

   a). Edit the following flags in `/opt/mellanox/doca/applications/meson_options.txt`:

   - Set `enable_all_applications` to `false`
   - Set `enable_nat` to `true`

   b). Run the commands in step 2.

> Note: NAT application is created under `./build/nat/src/`.

Application usage:

```
Usage: doca_nat [DPDK Flags] -- [DOCA Flags] [Program Flags]

DOCA Flags:
  -h, --help                        Print a help synopsis
  -v, --version                     Print program version information
  -l, --log-level                   Set the log level for the program
 <CRITICAL=20, ERROR=30, WARNING=40, INFO=50, DEBUG=60>

Program Flags:
  -m, --mode <mode>                 set NAT mode
  -r, --nat-rules <path>            Path to the JSON file with NAT rules
  -lan, --lan-intf <lan intf>       name of LAN interface
```

```
  -wan, --wan-intf <wan intf>       name of wan interface
```

Note: For additional information on available flags for DPDK, use `-h` before the `--` separator:

```
/opt/mellanox/doca/applications/nat/bin/doca_nat -h
```

Note: For additional information on the application, use `-h` after the `--` separator:

```
/opt/mellanox/doca/applications/nat/bin/doca_nat -- -h
```

4. Running the application on BlueField:

   a). The NAT example is based on DPDK libraries. Therefore, the user is required to provide DPDK flags and allocate huge pages:

   ```
   sudo echo 2048 > /sys/kernel/mm/hugepages/hugepages-2048kB/nr_hugepages
   ```

   b). CLI example for running the application:

   ```
   /opt/mellanox/doca/applications/nat/bin/doca_nat
    -a auxiliary:mlx5_core.sf.4,dv_flow_en=2 -a
    auxiliary:mlx5_core.sf.5,dv_flow_en=2 -- -m static -r /opt/mellanox/doca/
   applications/nat/bin/nat_static_rules.json -lan sf3 -wan sf4
   ```

   Note: The flag `-a auxiliary:mlx5_core.sf.4 -a auxiliary:mlx5_core.sf.5` is mandatory for proper usage of the application. Modifying this flag results in unexpected behavior as only 2 ports are supported. The SF number is arbitrary and configurable.

   Note: SFs must be enabled according to Scalable Function Setup Guide.

5. To run `doca_nat` using a JSON file:

   ```
   doca_nat --json [json_file]
   ```

   For example:

   ```
   cd /opt/mellanox/doca/applications/nat/bin
   ./doca_nat --json nat_params.json
   ```

Chapter 7. Arg Parser DOCA Flags

For more information, refer to NVIDIA DOCA Arg Parser Programming Guide.

Flag Type	Short Flag	Long Flag/ JSON Key	Description	JSON Content
DPDK Flags	a	devices	Add a PCIe device into the list of devices to probe	"devices": [{"device": "sf", "id": "4","h true}, {"device": "sf", "id": "5","h true},]
General Flags	l	log-level	Set the log level for the application: ▶ CRITICAL=20 ▶ ERROR=30 ▶ WARNING=40 ▶ INFO=50 ▶ DEBUG=60	"log-level": 60
	v	version	Print program version information	N/A
	h	help	Print a help synopsis	N/A
Program Flags	m	mode	Set NAT mode	"mode": "static"
	r	nat-rules	Path to the JSON file with NAT rules	"nat-rules": "nat_static_rules.jso
	lan	Lan-intf	Name of LAN interface	"lan-intf": "sf3"
	wan	Wan-intf	Name of WAN interface	"wan-intf": "sf4"

Chapter 8. References

- `/opt/mellanox/doca/applications/nat/src`

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

NVIDIA, the NVIDIA logo, and Mellanox are trademarks and/or registered trademarks of Mellanox Technologies Ltd. and/or NVIDIA Corporation in the U.S. and in other countries. The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a world¬wide basis. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

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