



# NVIDIA DOCA DNS Filter

Reference Application

# Table of Contents

- Chapter 1. Introduction..... 1
- Chapter 2. System Design..... 2
- Chapter 3. Application Architecture..... 5
- Chapter 4. Configuration Flow..... 6
- Chapter 5. Running Application on BlueField..... 8
- Chapter 6. Running Application on Host..... 10
- Chapter 7. References..... 11

---

# Chapter 1. Introduction

Domain name system (DNS) translates domain names to IP addresses so browsers can load internet resources. Each device connected to the internet has a unique IP address which other machines use to find the device.

The DNS process includes several steps:

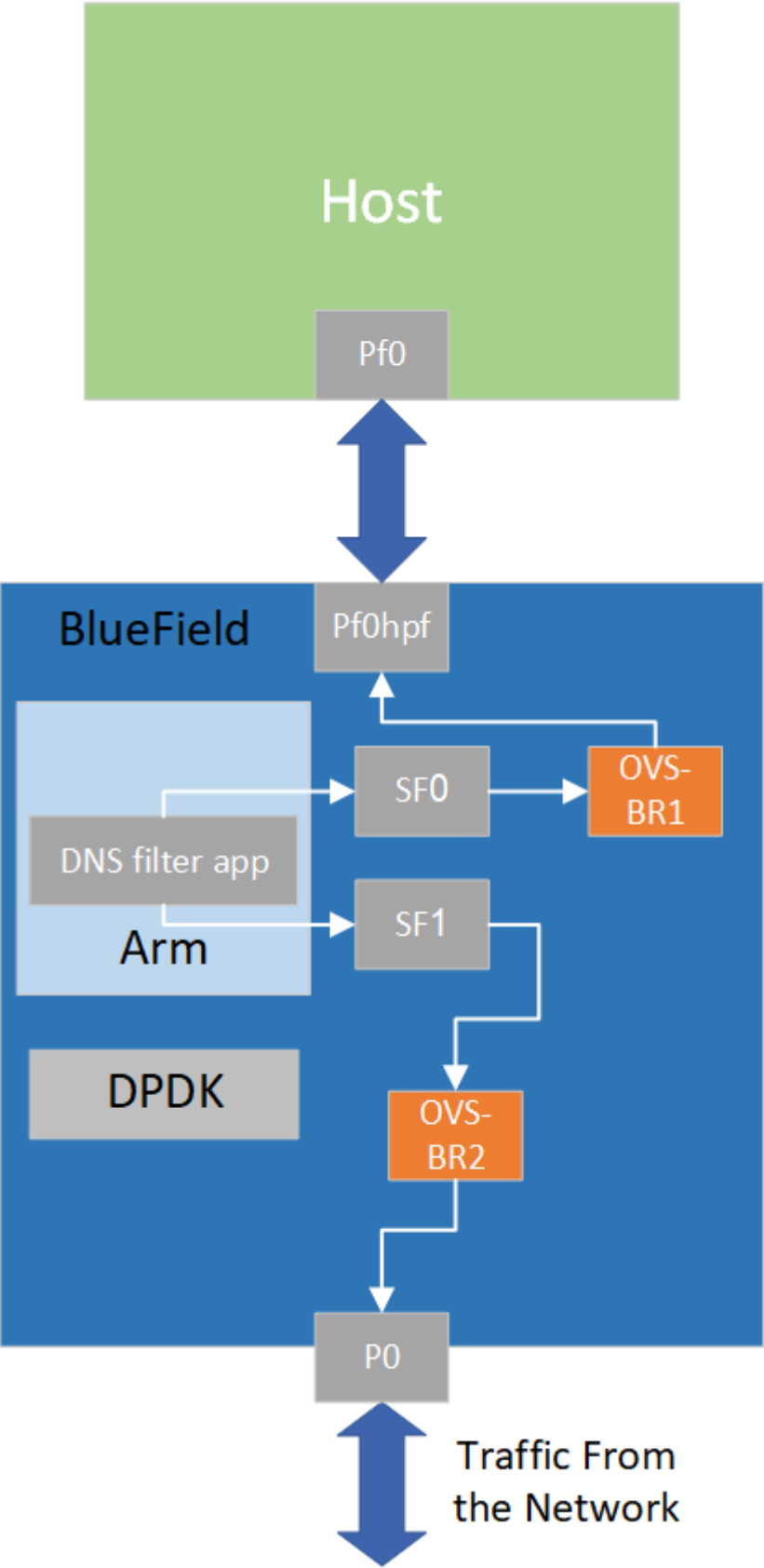
1. Once a user tries to log into a website using a browser, the user's device creates a DNS query and sends it to a DNS resolver.
2. The DNS resolver queries the DNS domain to get an IP address by searching its cache or sending the request to another DNS server.
3. Once a match is found, the DNS resolver returns the correct IP matching the DNS domain.
4. The user can log into the required website using the correct IP.

DNS filter is used to offload DNS requests from the host to the BlueField DPU Arm which allows reducing CPU overhead as Arm allows further DNS processing to be done (e.g., whitelisting, logging, filtering, etc).

---

## Chapter 2. System Design

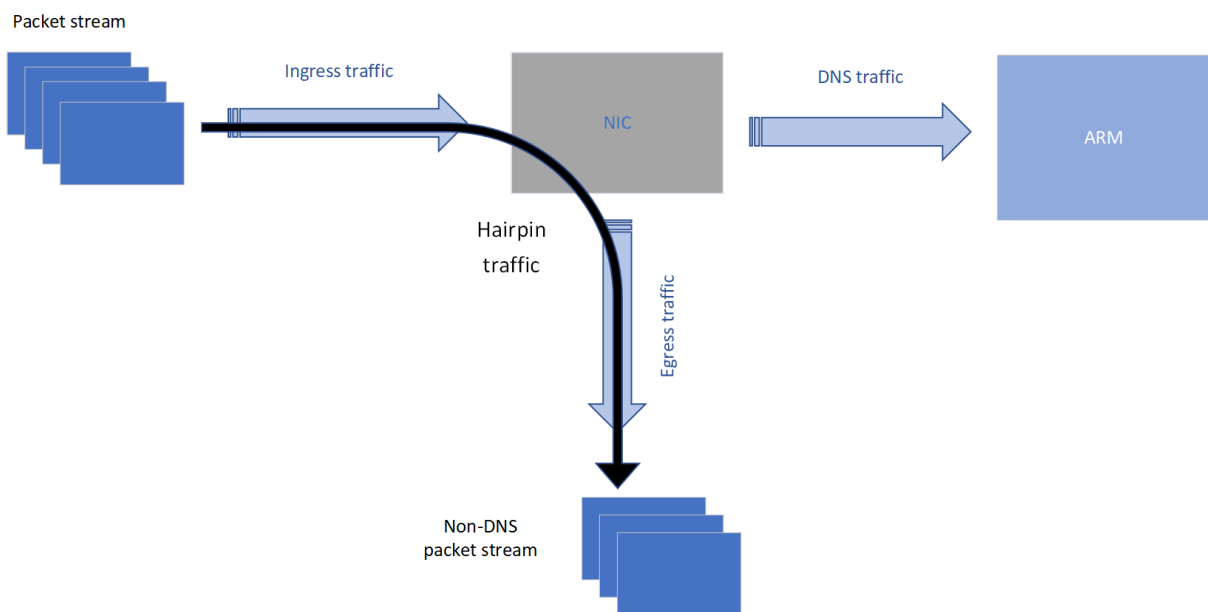
The DNS filter application is designed to run as a "bump-on-the-wire" on the BlueField-2 DPU instance. The DPU intercepts the traffic coming (ingress traffic) from the wire and either passes it to the Arm or forwards it to the egress port using hairpin. The decision is made by traffic classification.





# Chapter 3. Application Architecture

The DNS filter runs on top of DOCA FLOW to classify DNS requests.



1. Ingress packet types are identified using pipes which encapsulate flow rule matching patterns and actions.
2. Matched flows are identified, and FORWARDING actions can be executed.
  - ▶ DNS traffic is forwarded to the Arm for further processing
  - ▶ Non-DNS traffic is forwarded to the egress port using hairpin

---

# Chapter 4. Configuration Flow

1. DPDK initialization.

```
dpdk_init(&argc, &argv, &nb_queues, &nb_ports);
```

2. Stateful flow table (SFT) and port initialization.

```
dpdk_ports_init(nb_queues, nb_ports);
```

- ▶ Mempool allocation
- ▶ Rx/Tx and hairpin queue initialization
- ▶ DPDK port initialization

3. Hairpin binding.

```
enable_hairpin_queues(portid, &peer_ports, 1);
```

- ▶ Binds hairpin queues for the given port ID

4. DOCA flow initialization.

```
doca_flow_init(&dns_flow_cfg, &error);
```

5. DOCA flow ports initialization.

```
dns_filter_port_init(&port_cfg, portid);
```

- ▶ Initializes DOCA flow port with the given port configuration for the given port ID.



**Note:** DOCA flow port initialization is done for both ports of the BlueField and after the DPDK ports have been initialized.

6. Non-DNS hairpin traffic.

```
build_hairpin_pipes(ports[portid], portid, nb_queues);
```

- ▶ Builds two hairpin pipes, that forward packets to Arm. For a given port, each pipe has one entry for the relevant matching patterns. The first hairpin pipe is for matching UDP non-DNS traffic and the second one is for matching TCP traffic. Note that these pipes are built for both ports of the BlueField.

7. Build DNS pipe.

```
build_dns_pipes(ports[portid], portid, nb_queues);
```

- ▶ Builds DNS pipe for a given port. The built pipe has one entry for matching DNS traffic and forwarding it to Arm.

8. Processing packets.

```
main_loop(nb_queues, nb_ports);
```



- ▶ All received packets on Arm, are DNS packets, while non-DNS packets are forwarded to the egress port using hairpin allowing DNS packets to be filtered.

---

# Chapter 5. Running Application on BlueField

1. Please refer to the [DOCA Installation Guide](#) for details on how to install BlueField related software.
2. To build the application
  - a). The DNS filter example is installed as part of the `doca-dpi-lib` package, the binary is located under `/opt/mellanox/doca/examples/dns_filter/bin/doca_dns_filter`. To re-build the DNS filter sample, run:

```
cd /opt/mellanox/doca/examples/dns_filter/src
meson /tmp/build
ninja -C /tmp/build
```

`doca_dns_filter` will be created under `tmp/build`.

- b). The build process depends on the `PKG_CONFIG_PATH` environment variable to locate the DPDK libraries. If the variable was accidentally corrupted, and the build fails, run the following command:

- ▶ For Ubuntu:

```
export PKG_CONFIG_PATH=$PKG_CONFIG_PATH:/opt/mellanox/dpdk/lib/aarch64-linux-gnu/pkgconfig
```

- ▶ For CentOS:

```
export PKG_CONFIG_PATH=$PKG_CONFIG_PATH:/opt/mellanox/dpdk/lib64/pkgconfig
```

- c). The DNS filter example is a DPDK application. Therefore, the user is required to provide DPDK flags and allocate huge pages. Run:

```
echo 1024 > /sys/kernel/mm/hugepages/hugepages-2048kB/nr_hugepages
```

3. To run the application:

```
./doca_dns_filter [dpdk flags] -- -l [log_level]
```



**Note:** SFs must be enabled according to [Scalable Function Setup Guide](#).

For example:

```
/opt/mellanox/doca/examples/dns_filter/bin/doca_dns_filter -a
auxiliary:mlx5_core.sf.4 -a auxiliary:mlx5_core.sf.5 -- -l 3
```



**Note:** The flag `-a auxiliary:mlx5_core.sf.4 -a auxiliary:mlx5_core.sf.5` is a must for proper usage of the application. Modifying this flag will result unexpected behavior as only two ports are supported. The SF number is arbitrary and configurable. For additional information on available flags for DPDK, use `-h` before the `--` separator. For

information on available flags for the application, use `-h` after the `--` separator. The `-l` or `--log_level` flag sets the log level for the app (ERR=0, DEBUG=3).

---

## Chapter 6. Running Application on Host

Please refer to [Running Reference Applications Over Host Guide](#).

---

## Chapter 7. References

- ▶ `/opt/mellanox/doca/examples/dns_filter/src/dns_filter.c`

## 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 NON-INFRINGEMENT, 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. Other company and product names may be trademarks of the respective companies with which they are associated.

## Copyright

© 2021 NVIDIA Corporation & affiliates. All rights reserved.