NVIDIA DOCA Netflow

Programming Guide

MLNX-15-060474 _v1.1 | August 2021

Table of Contents

Chapter 1. Introduction 1
1.1. Related Documentation 1
1.2. Intended Audience 1
Chapter 2. DOCA NetFlow Library 2
2.1. Installation and Prerequisites 2
2.2. NetFlow API 3

Chapter 1. Introduction

DOCA Utils are utility libraries intendent to be used alongside other DOCA packages (such as DOCA DPI) to extend their capabilities.

1.1. Related Documentation

Document Name	Description
DPDK Installation	Details how to install and compile DPDK
NVIDIA BlueField DPU Family Software Documentation	This document provides product release notes as well as information that explains the BlueField Software Distribution (BSD) and how to develop and/or customize applications, system software, and file system images for the BlueField platform.

1.2. Intended Audience

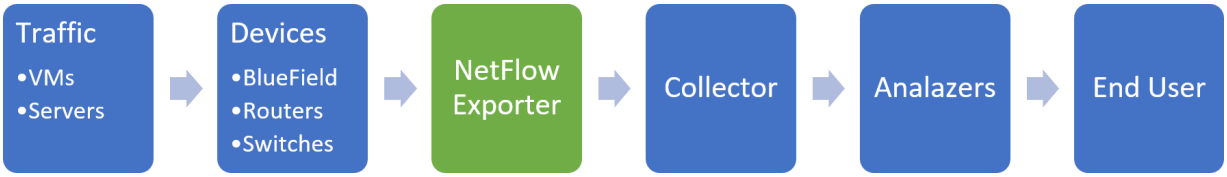
This document is intended for software developers writing DOCA-based applications and need to extend the app capabilities with utilities libraries such as export NetFlow format.

The document assumes familiarity with the DOCA packages and data plane development kit (DPDK).

Chapter 2. DOCA NetFlow Library

The NetFlow lib simplifies and centralizes the formatting and exporting of NetFlow packets.

NetFlow is a protocol invented by Cisco for exporting device network flow information to a NetFlow collector. The collector saves and aggregates data. From there, the analyzer processes the data and produces insights on the organization traffic.



Note: This lib only implements NetFlow exporter.

NetFlow protocol version 9 features custom template capabilities. With custom templates, the application can send any set of fields from a predefined field list.

The basic output of NetFlow is a flow record, the record contains info of one flow (e.g. 5-tuple, bytes count, packets count, application name, etc). For more information about flows, please refer to DOCA DPI Programming Guide. Each flow can have several records in the collector over time that differ in byte count, packet count, delivery time and so on.

2.1. Installation and Prerequisites

Note: Please refer to DOCA Installation Guide for installation requirements.

The NetFlow example is provided as part of the DOCA Utils library packages and is provided both for Debian based operating systems as well as RH. The NetFlow example will be installed into `{prefix}/bin`. The source will be installed into `{prefix}/src/doca/utils/netflow`.

The configuration file is installed into `/etc/doca-netflow.conf`. The configuration file can be placed anywhere but this is the default path.

The configuration file should contain:

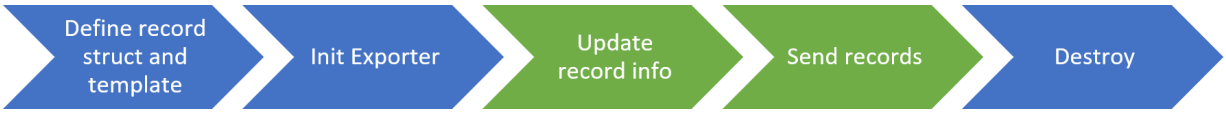
- Title – "[doca_netflow_conf]"
- Target – the IP address and port of the collector (where to send the NetFlow records)

- Source ID – the source ID of the NetFlow exporter (defined by the end user) to be sent alongside NetFlow records to identify the source of the record
- Version – 9 (for future development)

For example:

```
[doca_netflow_conf]
target = 127.0.0.1:2055
source_id = 10
version = 9
```

2.2. NetFlow API



1. Define record struct and template.

   The app must define a TLV NetFlow v9 template. This is done by using `doca_netflow_types.h` defined macros. The order has no significance for the collector but must be maintained for the record struct. The struct must be in packed format without pointers.

   The app can use the default record struct and template by using structs `doca_netflow_default_record` and `doca_netflow_template_default_get()`.

   ```
   struct record_exmaple {
   uint32_t src_addr_V4;
   uint32_t dst_addr_V4;
   }

   struct doca_netflow_flowset_field fields[] = {
   {.type = DOCA_NETFLOW_IPV4_SRC_ADDR, .length =
    DOCA_NETFLOW_IPV4_SRC_ADDR_DEFAULT_LENGTH},
   {.type = DOCA_NETFLOW_IPV4_DST_ADDR, .length =
    DOCA_NETFLOW_IPV4_DST_ADDR_DEFAULT_LENGTH}
   };

   struct doca_netflow_template template = {
   .field_count = 2;
   .fields = fields;
   };
   ```

   Alternatively, users may use the default template:

   ```
   struct doca_netflow_template  *template = doca_netflow_template_default_get();
   ```

2. Initialize NetFlow exporter.

   Call `doca_netflow_exporter_init()` to initiate the exporter. Pass the configuration file or pass NULL for the default path.

   ```
   doca_netflow_exporter_init("path/to/doca_netflow.conf");
   ```

   Or:

   ```
   doca_netflow_exporter_init(NULL); //the default location is used
   ```

3. Update record information.

The app must update the flow information with the process that extracts it from the packets. For information on said process, please refer to the NVIDIA DOCA Application Recognition Reference Guide.

This part must be periodically updated before sending the records.

```
struct record_exmaple record = {
    .src_addr_v4 = inet_network("192.168.120.1"),
    .dst_addr_v4 = inet_network("192.168.120.2"),
};
records[i] = &record;
```

4. Send records.

   Call `doca_netflow_exporter_send()` to send the records to the collector.

   This function should be periodically called as it contains the records of all the current active flows or flows sampled in that period time.

   ```
   doca_netflow_exporter_send(template, (const void **)(records), records_len,
    &err);
   ```

5. Destroy exporter.

   Call `doca_netflow_exporter_destroy()` to close the exporter before exiting the app.

   ```
   doca_netflow_exporter_destroy();
   ```

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. 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.

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