napatech

Quick Guide

Napatech Link™ Capture Software for Intel® PAC with Intel® Arria® 10 GX FPGA

Installation and Use

Use this information to install and configure Napatech Link™ Capture Software for Intel® PAC with Intel® Arria® 10 GX FPGA.

DN-1113 Rev. 11

Modification History

This document has been updated as follows:

Rev.	Date	Comment
1	2018-06-28	Initial version.
2	2018-08-31	First release.
3	2018-11-09	• Overview image updated: Overview on page 7 • Various installation updates: Installing on page 8, Set up BIOS and Kernel Options on page 10, Installing the OPAE Framework on page 12, Installing Napatech Link™ Capture Software on page 13, Selecting 1×40G or 4×10G on page 15, and Requesting the License Key for Napatech Link™ Capture Software on page 17 • Update to Suricata topic: Suricata on page 46
4	2018-11-30	Changes have been made in Section 1.1. A number of minor changes have been made.
5	2018-12-13	Changes have been made in Section 2.5. A number of minor changes have been made.
6	2018-12-17	Various changes have been made.
7	2019-03-14	Updates to serial number instructions: Requesting the License Key for Napatech Link™ Capture Software on page 17. A number of other minor changes have been made.
8	2019-03-19	Some minor changes have been made.
9	2019-03-21	Updates to Accelerator Stack guidance: Installing the OPAE Framework on page 12
10	2019-04-05	• Split out requesting and installing license key: Requesting the License Key for Napatech Link™ Capture Software on page 17 and Installing the License Key for Napatech Link™ Capture Software on page 19 • Clarification of network visibility: Installing Napatech Link™ Capture Software on page 13 and Overview on page 7
11	2019-06-26	Small edits.

Contents

1 Introduction

Target audience

This guide is aimed at system administrators at enterprise data centers and system integrators working for enterprise data centers.

You need a working knowledge of package-based installation of software and basic configuration of a Linux server.

Terminology and trademarks

Intel, **Arria** and **Xeon** are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

In this document, the terms "**Intel® PAC A10 GX** acceleration card" or **Intel® PAC with Intel® Arria® 10 GX FPGA** are used to denote **Intel® Programmable Acceleration Card with Intel® Arria® 10 GX FPGA**.

This chapter contains the following sections:

1.1 Release Information

This release

This release contains a number of product packages. Two are relevant for the Intel® PAC with Intel® Arria® 10 GX FPGA and this document:

- **ntanl_package_3gd_intel_pac-x.y.z-linux**

  If you have downloaded this package, please refer to The dedicated intel_pac product package on page 6 for information about the package.

- **ntanl_package_3gd-x.y.z-linux**

  If you have downloaded this package, please refer to the relevant Release Summary document in the package for information about the release and the package.

The dedicated intel_pac product package

The **ntanl_package_3gd_intel_pac-x.y.z-linux** product package includes two Accelerator Functional Units (AFUs) for the Intel® PAC with Intel® Arria® 10 GX FPGA for running at 1 × 40 Gbit/s and at 4 × 10 Gbit/s, respectively, Napatech Software Suite, tools packages, libpcap and documentation.

The package supports Linux kernel 3.10 to 4.7, 64-bit for Intel® PAC with Intel® Arria® 10 GX FPGA.

The included documents are:

- Napatech, Napatech Link™ Capture Software for Intel® PAC with Intel® Arria® 10 GX FPGA, Installation and Use, Quick Guide, DN-1113 (this document)
- Napatech, Napatech Software Suite, Reference Documentation, DN-0449

These documents are sufficient for standard use of Napatech Link™ Capture Software with the Intel® PAC with Intel® Arria® 10 GX FPGA.

Additional documentation relevant for more advanced use of Napatech Link™ Capture Software with the Intel® PAC with Intel® Arria® 10 GX FPGA can be found on docs.napatech.com (the Napatech Documentation Portal) by selecting the **Intel® PAC** platform (see Intel® PAC documentation).

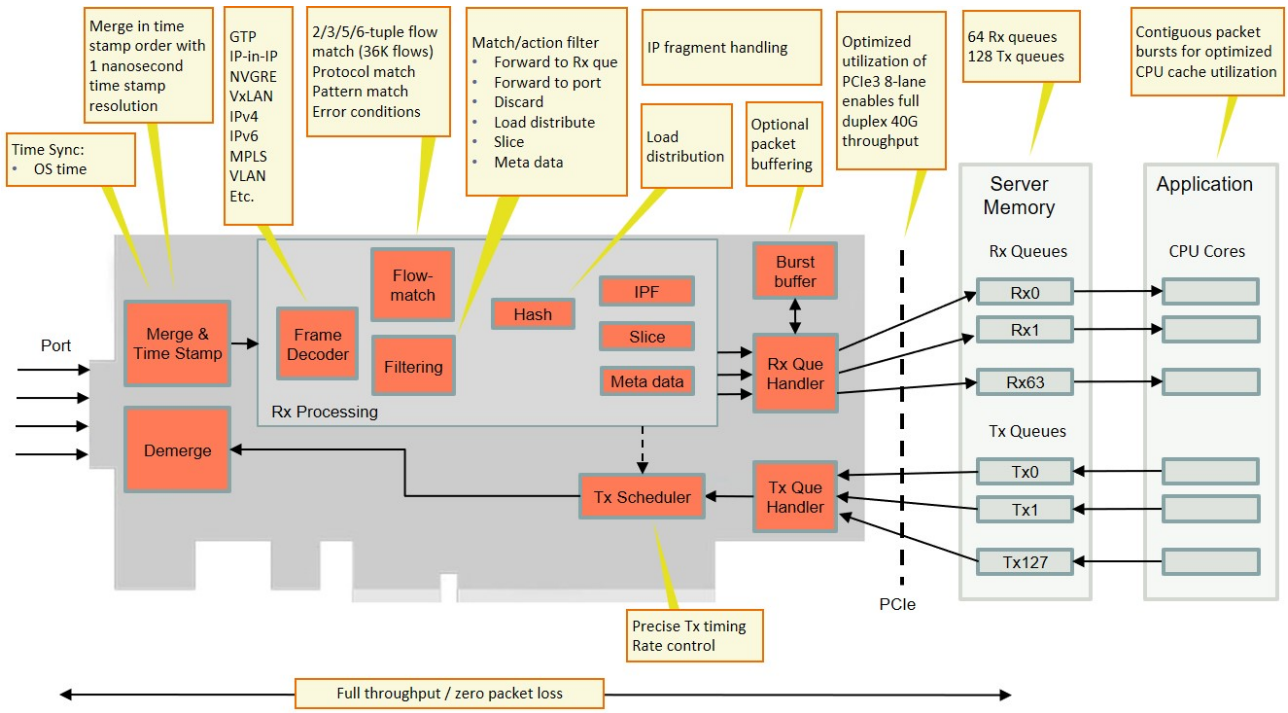
1.2 Overview

Napatech Link™ Capture Software with Intel® PAC A10 GX passively captures traffic on a network, and uses a mirror port or span port to connect to the network. This ensures that the solution can provide zero packet loss, and is not detectable on the network. As such, Napatech Link™ Capture Software with Intel® PAC A10 GX does not show up as a standard network interface when you run network tools such as **`ifconfig`** or **`ip`**.

Features of Napatech Link™ Capture Software with Intel® PAC A10 GX

The feature set offloads processing and analysis of Ethernet data from application software while ensuring optimal use of standard server resources, leading to effective application acceleration. Capabilities include:

- Full line-rate packet capture and transmit
- Optimized packet buffers to minimize CPU load on the host server
- Onboard packet buffering to ensure packet delivery during congestion
- HW time-stamping with 1 ns time stamp resolution



For a full description of the feature set, see DN-1128.

2 Installing

There are a number of prerequisites for installing Napatech Link™ Capture Software for use with Intel® PAC A10 GX. You can then prepare the server for installation, and install the Napatech Link™ Capture Software.

Prerequisites

Ensure that:

1. Your Intel platform is from the Intel® Xeon® Processor E5 Family or higher. This is because we require support for PCIe Access Control Services (ACS).
2. You have installed RHEL/CentOS 7.4 or later.

   Use the `lsb_release` command to check the operating system release.

   Install `lsb_release` and check your release version:

   ```
   sudo yum install redhat-lsb
   lsb_release -d
   ```

   You will see output similar to this:

   ```
   Description: CentOS Linux release 7.5.1804 (Core)
   ```

3. You have installed one or more Intel® PAC A10 GX acceleration cards.

   Use the `lscpi` command to list the installed Intel® PAC A10 GX acceleration cards and Napatech SmartNICs.

   Install `pciutils` and check the installed cards:

   ```
   sudo yum install pciutils
   lspci -nn |grep -E '(8086:09c4|18f4:)'
   ```

   You will see output similar to this:

   ```
   04:00.0 Processing accelerators [1200]: Intel Corporation Device
   [8086:09c4]
   82:00.0 Network controller [0280]: Napatech A/S NT40E3-4-PTP
   Network Adapter 4x10Gb [18f4:0145]
   ```

4. You have installed the QSFP+ 40G pluggables, optionally with breakout cables to 4×10G, in each Intel® PAC A10 GX acceleration card.

Preparing the server for installation of Napatech Link™ Capture Software

Complete these tasks to prepare the server:

1. Set BIOS and kernel options: Set up BIOS and Kernel Options on page 10
2. Install the OPAE framework: Installing the OPAE Framework on page 12

Installing Napatech Link™ Capture Software

Complete these tasks to install the software:

1. Install Napatech Link™ Capture Software: Installing Napatech Link™ Capture Software on page 13
2. Select 4×10G or 1×40G loadable AFU image: Selecting 1×40G or 4×10G on page 15
3. Request license keys: Requesting the License Key for Napatech Link™ Capture Software on page 17
4. Install license keys: Installing the License Key for Napatech Link™ Capture Software on page 19

This chapter contains the following sections:

2.1 Set up BIOS and Kernel Options

Before installing and running Napatech Link™ Capture Software with Intel® PAC A10 GX, you must set some BIOS and kernel options.

1. Ensure you have met the installation prerequisites in Prerequisites on page 8.
2. Ensure that virtualization is enabled in the BIOS, by enabling these settings:
   - Intel VT-d (Intel Virtualization Technology for Directed I/O)
   - Intel VT-x (Intel Virtualization Technology for IA-32 and Intel 64 Processors)
3. In the `/etc/default/grub` file, in the line starting `GRUB_CMDLINE_LINUX=`, add this code:

```
intel_iommu=on
```

4. (Optional) If you want to run the DPDK application, enable the Linux kernel option `hugepages`.

   In the `/etc/default/grub` file, in the line starting `GRUB_CMDLINE_LINUX=`, add this code:

```
default_hugepagesz=1G hugepagesz=1G hugepages=2
```

5. Update the kernel options:

   If you are using UEFI-based boot, the file `/sys/firmware/efi` exists. Run these commands:

```
sudo su
grub2-mkconfig -o $(readlink -e /etc/grub2-efi.cfg)
```

   Otherwise, run these commands:

```
sudo su
grub2-mkconfig -o $(readlink -e /etc/grub2.cfg)
```

6. Reboot the server:

```
shutdown -r now
```

7. Confirm your kernel options by running this command:

```
cat /proc/cmdline
BOOT_IMAGE=/vmlinuz-3.10.0-862.6.3.el7.x86_64
root=/dev/mapper/centos-root ro crashkernel=auto
rd.lvm.lv=centos/root rd.lvm.lv=centos/swap rhgb quiet
intel_iommu=on
```

8. Confirm that IOMMU and VT-d are enabled by running this command:

```
$ dmesg |grep DMAR
[    0.000000] ACPI: DMAR 000000007b6fe000 00118 (v01 DELL   PE_SC3
   00000001 DELL 00000001)
[    0.000000] DMAR: IOMMU enabled
...
[    1.185400] DMAR: Intel(R) Virtualization Technology for
Directed I/O
```

You are now ready to install the OPAE framework.

2.2 Installing the OPAE Framework

Install the Intel Acceleration Stack release as a base for the Napatech software.

The Intel Acceleration Stack comprises a large number of frameworks and tools for developing and testing Accelerator Functions (AFs). We require only the OPAE Intel FPGA driver, and the OPAE tools, however we recommend that you download the entire stack, using Intel's instructions.

Before you begin

Make sure you have read your Napatech release note to find out what version of the Intel Accelerator Stack software you require.

You need to install the following:

- Linux kernel development headers:

```
sudo yum install kernel-devel-$(uname -r)
```

- GCC version 4.0 or later, **`glibc`** and **`make`**:

```
sudo yum groupinstall 'development tools'
```

- DKMS:

```
sudo yum install --enablerepo=extras epel-release
sudo yum install dkms
```

- Other dependencies of the OPAE framework:

```
sudo yum install json-c uuid libuuid
```

Procedure

1. Open the Intel Downloads page: Downloads for Intel FPGA Acceleration Hub.

   You need a "My Intel" account to log in and download an Intel Acceleration Stack release package.

2. Scroll to **Acceleration Stack for Runtime**, choose the Accelerator Stack version and click **Download Now**.

   Check your Napatech software release note for information on the version of Intel Acceleration Stack software that you require.

3. Install the Intel Acceleration Stack for Runtime: Installing the Intel Acceleration Stack Runtime package on the Host Machine.

   Answer **Yes** when prompted: *Do you wish to install the OPAE?*

4. Verify the FIM version: Identifying and Updating the FIM.

   If you need to update the FIM, see also the next topic: Updating Flash using the fpgaflash Tool.

   Note: If you update the FIM, you will need access to power cycle the server.

2.3 Installing Napatech Link™ Capture Software

Install Napatech Link™ Capture Software using the package installer script included in the product release package.

Prerequisite

You need to install the following:

- Linux kernel development headers, GCC version 4.0 or later, **`glibc`**, and **`make`** (you should have already installed these in Installing the OPAE Framework on page 12)
- ncurses version 5.0 or later (only applies to the `monitoring`, `profiling` and `setup_wizard` tools):

```
sudo yum install ncurses
```

Step	Action
1	Go to the Napatech site: https://www.napatech.com/products/link-capture-software-for-intel/.
2	From the toolbar, click **Download Software**.  Scroll down to **Link™ Capture Software for Intel® Programmable Acceleration Card with Intel Arria® 10 GX FPGA** and download either the Windows or Linux product package.

3	Unpack the product package:
	`tar xzf ntanl_package_3gd_intel_pac-x.y.z-linux.tar.gz`
	`cd ntanl_package_3gd_intel_pac-x.y.z-linux`
	Step result:
	The unpacked product directory contains:
	• Product package release notes.
	• A readme file, describing the product package content.
	• A package install shell script, a product package install script.
	• A `documentation` subdirectory.
	• A subdirectory for each product package component, containing release notes and an installer file for the component.

4	Install the Napatech driver and tools: **Note:** You need root privileges to install and compile the Linux driver and the tools. `sudo package_install_3gd-x.y.z.sh` The product package installer script is located in the root directory. This installs the driver software, the tools, documentation, FPGA images, and optionally a version of libpcap. If the product package installer script detects that a version of the driver software is already installed, you can update the existing installation, delete the existing installation before commencing, or quit the installation script. You can install a version of libpcap with Napatech extensions. For more information about libpcap with Napatech extensions, see DN-0428. The `/opt/napatech3/` directory, including subfolders, is created. The driver is installed and compiled to match the target Linux kernel version. **Note:** Napatech Link™ Capture Software with Intel® PAC A10 GX does not show up as a standard network interface when you run network tools such as **`ifconfig`** or **`ip`**. See Overview on page 7 for more information.

2.4 Selecting 1×40G or 4×10G

Napatech Link™ Capture Software offers two different port configurations, 4×10G and 1×40G, for Intel® PAC A10 GX.

AFU image files

The firmware part of Napatech Link™ Capture Software is packaged as FPGA image files. FPGA image files for Intel® PAC A10 GX are known as AFU (Accelerator Functional Unit) image files. AFU image files can be dynamically loaded to Intel® PAC A10 GX acceleration cards.

Napatech Link™ Capture Software offers these AFU image files for Intel® PAC A10 GX:

Port configuration	AFU filename
4×10G	200-7000-xx-xx-xx-xxxxxx-xxxx.egbs
1×40G	200-7001-xx-xx-xx-xxxxxx-xxxx.egbs

AFU image files are also known as Green BitStream files and commonly use `.gbs` as the filename extension. AFU image files for use with Napatech Link™ Capture Software are encrypted and use `.egbs` as the filename extension.

During initialization, **`ntservice`** reads `/opt/napatech3/config/afu.ini` and `/opt/napatech3/config/ntlicense.key` and loads the specified encrypted AFU image files. Your license key is valid for both types of port configuration. See Installing the License Key for Napatech Link™ Capture Software on page 19 for information about how to order and install a license key file.

Selecting 4×10G or 1×40G

1. Backup any previous version of the `/opt/napatech3/config/afu.ini` file.

   The installation will overwrite any previous versions.

2. Run the `/opt/napatech3/bin/nt_afu_ini_setup.sh` configuration script to select an AFU image file for each installed Intel® PAC A10 GX:

   ```
   sudo /opt/napatech3/bin/nt_afu_ini_setup.sh
   ```

3. If `/opt/napatech3/config/afu.ini` already exists, it is displayed. Type `Y` to confirm reconfiguration.

   ```
   OK: 1x40G image found
   /opt/napatech3/images/INTEL-A10-1x40/200-7001-12-03-00-180817-1210.egbs

   >>  Configure AFU images for PCI device found in PCI-slot
   0000:04:00.0 - ([Y]es/[N]o) ?
   ```

4. If `/opt/napatech3/config/afu.ini` does not exist, or if you confirm reconfiguration, the script displays bus IDs for the installed Intel® PAC A10 GX acceleration cards and filenames for available AFU image files.

   For each Intel® PAC A10 GX, choose whether to select another AFU image file and which one:

   ```
   >>  Configure AFU images for PCI device found in PCI-slot
   0000:04:00.0 - ([Y]es/[N]o) ?

   >>  select 4x10G (press 1) or 1x40G (press 2)
   ```

   Here is a sample of the commands and output:

```
$ sudo /opt/napatech3/bin/nt_afu_ini_setup.sh
No afu image configuration file (/opt/napatech3/config/afu.ini) found

Intel(R) PAC A10 PCI adapter(s) found in following PCI slots

---------------------
 No  |   PCI slot
---------------------
 0       0000:04:00.0

OK: 4x10G image found
/opt/napatech3/images/INTEL-A10-4x10/200-7000-12-02-00-180815-0828.egbs
OK: 1x40G image found
/opt/napatech3/images/INTEL-A10-1x40/200-7001-12-03-00-180817-1210.egbs

>>  Configure AFU images for PCI device found in PCI-slot 0000:04:00.0 - ([Y]es/[N]o) ?

>>  select 4x10G (press 1) or 1x40G (press 2)

OK: 4x10G selected

OK: afu configuration file, /opt/napatech3/config/afu.ini, created
```

Example `afu.ini` file

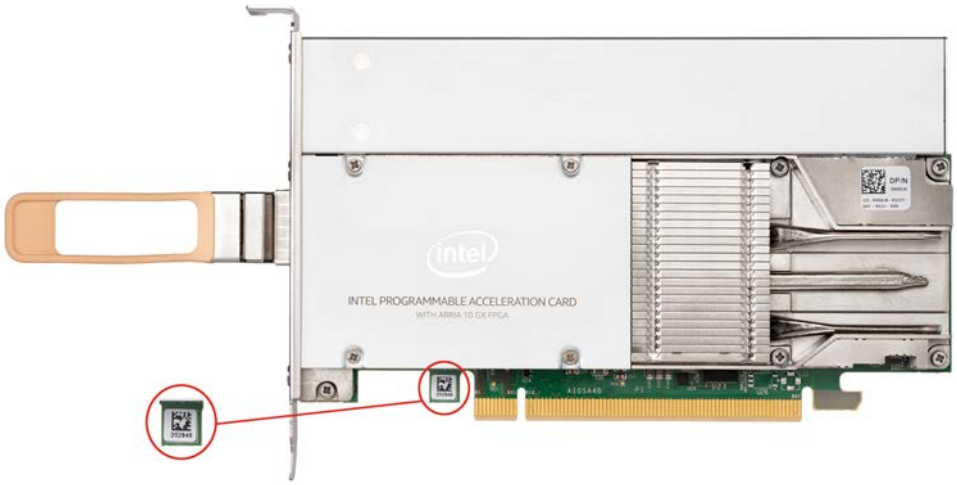
In this example `/opt/napatech3/config/afu.ini` file, the 4×10G AFU has been selected for the Intel® PAC A10 GX at BusID `0000:04:00.0`:

```
# afu.ini
#

[Afu0]
BusId = 0000:04:00.0
# 4x10G:
AfuFile0 = /opt/napatech3/images/INTEL-A10-4x10/200-7000-12-02-00-180815-0828.egbs
# 1x40G:
#AfuFile0 = /opt/napatech3/images/INTEL-A10-1x40/200-7001-12-03-00-180817-1210.egbs


#
# EOF
#
```

2.5 Requesting the License Key for Napatech Link™ Capture Software

Each Intel® PAC A10 GX acceleration card has a unique six-digit serial number, that relates to a specific license key. You will need this serial number when you request the license key.

1. If you have physical access to the Intel® PAC A10 GX acceleration card, you can find the six-digit serial number on a small tab close to the PCIe connector:



   In this example, the Intel® PAC A10 GX acceleration card has the serial number `202946`.

   Note: Make sure that you locate the number on the top face of the card, as shown in the image.

2. If you do not have physical access to the card, use the following tool from the Intel extra tools package to read the serial number. For example:

```
install_dir/opae-1.1.2-1/tools/extra/pac/pac_hssi_config/pac_hssi_config.py
 eeprom
```

   Here is an example of the output. The Intel® PAC A10 GX card is shown as serial number `202946`:

```
EEPROM:
```

```
(0x0000) 4D 41 43 3D 30 30 3A 30 62 3A 33 65 3A 30 31 3A MAC=00:0b:3e:01:
(0x0010) 61 61 3A 32 63 0A 53 4E 3D 32 30 32 31 34 37 0A aa:2c.SN=202946.
(0x0020) 50 43 3D 41 31 30 53 41 34 2D 30 55 2D 42 31 31 PC=A10SA4-0U-B11
(0x0030) 35 58 32 45 32 51 2D 32 32 2D 49 34 30 31 34 30 5X2E2Q-22-I40140
(0x0040) 52 2D 36 0A 52 45 56 3D 31 2E 38 2E 30 2E 30 2E R-6.REV=1.8.0.0.
(0x0050) 30 0A 0A FF FF FF FF FF FF FF FF FF FF FF FF 0...............
```

Note: If you do not have the Intel extra tools package installed, you can download it as part of the Intel Acceleration Stack for Runtime from the Intel Downloads page: Downloads for Intel FPGA Acceleration Hub.

You need a “My Intel” account to log in and download the package.

3. Go to the Napatech site to request your license key: https://www.napatech.com/products/link-capture-software-for-intel/how-to-buy/.

   Choose one of the following options:

   1. If you are requesting up to five licenses, click **BUY ON WEBSTORE**, and select the type of license that you want (regular or 30-day free trial).

      Select the number of licenses, and **ADD TO CART**, and **Proceed to checkout**.

   2. If you want more than five licenses, click **PURCHASE FORM**.

   Complete the web form to provide contact (and payment) details and the six-digit Intel® PAC A10 GX serial number for each card.

Alternatively, you can send an email to info@napatech.com, specifying your contact details and information about the order. You will be contacted by a sales representative who will guide you through the purchase order process.

4. Your license key or keys will be emailed to you.

2.6 Installing the License Key for Napatech Link™ Capture Software

When you receive your license key, follow these steps to install it.

Make sure you have requested and received your license key for each Intel® PAC A10 GX card. See Requesting the License Key for Napatech Link™ Capture Software on page 17 for more information.

1. Make a backup copy of the file.

   Note: This is in case you delete the entire installation directory `/opt/napatech3` during re-installation or update of the Napatech Link™ Capture Software.

2. Install the `ntlicense.key` file:

   ```
   sudo cp ntlicense.key /opt/napatech3/config/
   ```

   `ntlicense.key` is a text file that contains one or more license keys. It can contain license keys for several AFU images for the same Intel® PAC A10 GX acceleration card.

3. Verify the installation:

   a. Stop **`ntservice`** and remove the `/opt/napatech3/config/ntservice.ini` file:

   ```
   $ sudo /opt/napatech3/bin/ntstop.sh
   Stopping NTService (this may take a while)
   NTService stopped
   [Done]

   $ sudo rm /opt/napatech3/config/ntservice.ini
   ```

   b. Restart **`ntservice`** without any previous configuration to check that it now loads the selected AFU images during initialization:

   ```
   sudo /opt/napatech3/bin/ntstart.sh
   ```

   c. Check that the Intel® PAC A10 GX acceleration card is available by running this command:

   ```
   /opt/napatech3/bin/adapterinfo
   ```

   Here is an example of the output:

   ```
   adapterinfo (v. 3.9.1.133-1fad3)
   ================================================================================
   System: 3.9.1.133
   --------------------------------------------------------------------------------
   Adapters:  2
   Ports:     8
   ================================================================================
   Adapter 0: NT40E3-4-PTP-ANL Analysis Network Adapter PCIe Gen3 4x10Gb SFP+
   --------------------------------------------------------------------------------
   Bus ID:             0000:82:00.0
   PCI ID:             18F4:0145
   FPGA ID:            200-9502-10-07-00
   Feature level:      N-ANL11
   Profile:            Capture
   Supported profiles: Capture

   AVR version:  3.0i

   Ports:            4
   Port Range:       0 - 3
   Descriptor:       Standard
   Timestamp:        Native Unix
   TimestampMethod:  EOF (End of Frame)

   RX host buffers:   4
   TX host buffers:   4

   Port #0 MAC : 00.0D.E9.03.C2.48
   Port #1 MAC : 00.0D.E9.03.C2.49
   Port #2 MAC : 00.0D.E9.03.C2.4A
   Port #3 MAC : 00.0D.E9.03.C2.4B
   ```

```
SDRAM size  : 4GB

Bus type                : PCI Express 3
Link width supported    : x8
Link width negotiated   : x8
Link speed              : 8.0 GT/s
Max payload supported   : 512
Max payload negotiated  : 256
Max expected throughput : 58661 Mbps (calculated)
================================================================================
Adapter 1: Intel PAC A10 GX 4x10 Analysis Network Adapter
--------------------------------------------------------------------------------
Bus ID:             0000:04:00.0
PCI ID:             8086:09C4
FPGA ID:            200-7000-12-02-00
Feature level:      N-ANL12
Profile:            Capture
Supported profiles: Capture


Ports:            4
Port Range:       4 - 7
Descriptor:       Standard
Timestamp:        Native Unix
TimestampMethod:  EOF (End of Frame)

RX host buffers:   4
TX host buffers:   4

Port #4 MAC : 00.0B.3E.01.DD.AC
Port #5 MAC : 00.0B.3E.01.DD.AD
Port #6 MAC : 00.0B.3E.01.DD.AE
Port #7 MAC : 00.0B.3E.01.DD.AF

SDRAM size  : 4GB

Bus type                : PCI Express 3
Link width supported    : x8
Link width negotiated   : x8
Link speed              : 8.0 GT/s
Max payload supported   : 512
Max payload negotiated  : 256
Max expected throughput : 44914 Mbps
================================================================================
```

4. Your installation is complete.

3 Configuring Napatech Link™ Capture Software

Use the **setup_wizard** tool to configure Napatech Link™ Capture Software for applications that can use libpcap or NTAPI.

Introduction

You can configure Napatech Link™ Capture Software in several ways:

- Use the interactive **setup_wizard** tool that walks you through the configuration. For most users, this is the primary configuration method.
- To offload processing from your host CPUs, you can take advantage of Napatech hardware-accelerated filtering and packet processing. Do this by adding filter expressions to one of the configuration files created using the **setup_wizard** tool.

Filter expressions written in NTPL (Napatech Programming Language) can, for example, be used to hardware-accelerate filtering on SmartNIC ports, IP addresses, protocols and errored packets.

This chapter contains the following sections:

- Using the Setup Wizard on page 22

3.1 Using the Setup Wizard

The interactive **setup_wizard** tool walks you through the configuration and restarts **ntservice** when you save the configuration.

Introduction

The setup wizard allows you to:

- Map SmartNIC ports to libpcap interfaces and NTPL streams, merging and load-distributing traffic from the SmartNIC ports to the libpcap interfaces and NTPL streams.
- Set up transmit capabilities for libpcap interfaces.
- Select the algorithm used for load distribution. You can, for example, set up load distribution so that each data flow (traffic between a set of IP addresses) appears on one libpcap interface.
- Change various settings such as synchronization.

Prerequisites

You need privileges as root to run the **setup_wizard** tool.

Running the Setup Wizard tool

Start the **setup_wizard** tool from the command line:

```
sudo /opt/napatech3/bin/setup_wizard
```

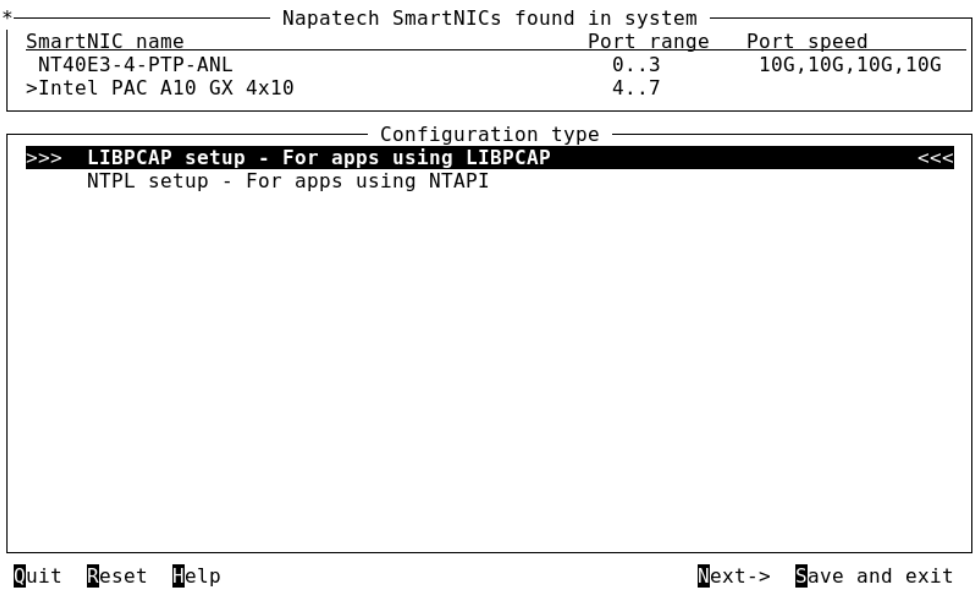
setup_wizard attempts to start **ntservice** if it is not already started:

```
Napatech driver does not appear to be started.
-> Starting Napatech driver.

Starting NTService (this may take a while)                    [Done]
```

setup_wizard restarts **ntservice** to activate the new configuration when you save and exit.

```
Stopping NTService (this may take a while)
NTService stopped                                             [Done]
Starting NTService (this may take a while)                    [Done]

The Napatech driver has been reconfigured and restarted successfully.
```

The Setup Wizard user interface



On pages with several groups of fields, the **Tab** key moves the focus *between* the groups. The active group is indicated by a '*' in the upper left corner.

The **Left**, **Right**, **Up**, and **Down** keys move the focus *within* a group of fields.

Press the **Enter** or **Space** key to select an option. In some cases, the focus moves the next group of fields.

Press the **Enter** or **Space** key to open an editable field. Press the **Enter** key or any navigation key to save the change and close the field. Press the **Esc** key to cancel changes and close the field.

Available commands are displayed in the bottom area. The shortcut key that runs the command is highlighted.

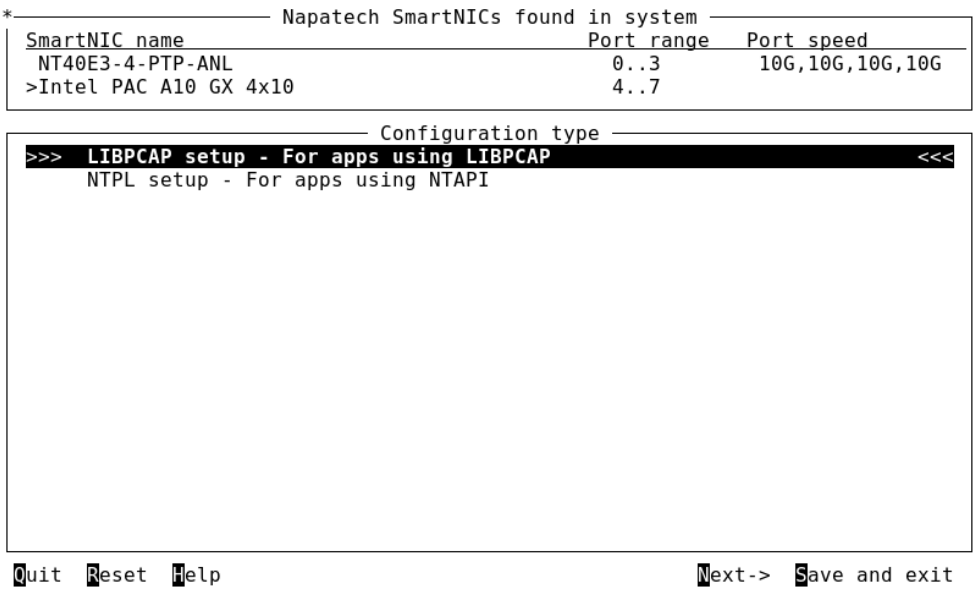
Some commands are common for all pages in the **`setup_wizard`** tool:

Command	Action
Quit	Quits the configuration without saving changes and without restarting **`ntservice`**.
Reset	Resets all pages to the default configuration. See Default Configuration on page 61 for more information.
Help	Displays help text.
Prev	Displays previous page.
Next	Displays next page.
Save and exit	Saves the configuration and restarts **`ntservice`**.

3.1.1 SmartNICs and Port Speeds

SmartNICs found in system

The upper area displays a list of available SmartNICs and the port numbering. You can edit the port speed for dual-rate pluggable modules.

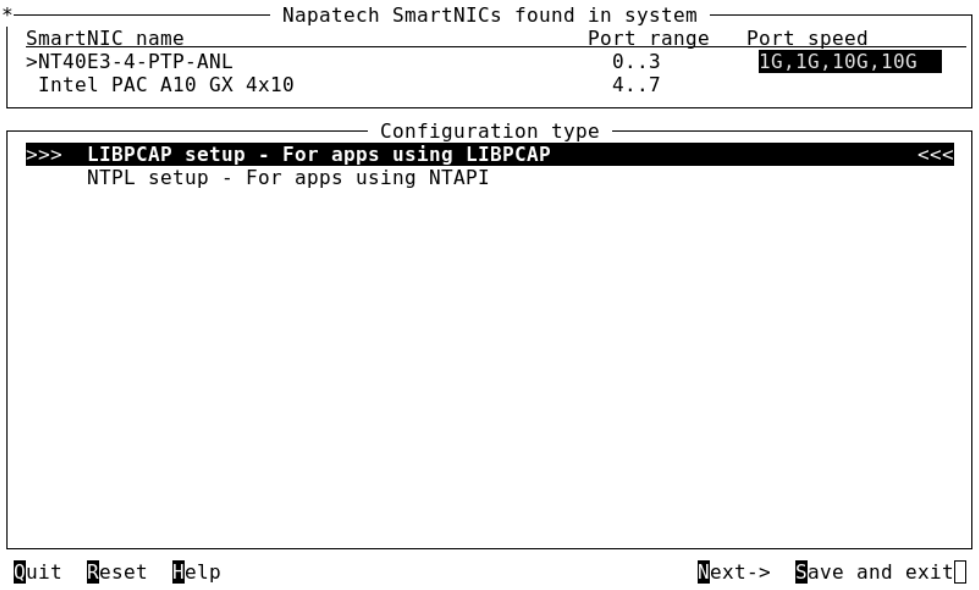


Port speed for dual-rate pluggable modules

Press the **Tab** key to move the focus to the SmartNIC list.

Press **Up** or **Down** to move the focus between SmartNICs.

Edit the **Port speed** field to select the port speed for 1/10Gbit/s dual-rate pluggable modules.



3.1.2 Configuration Type

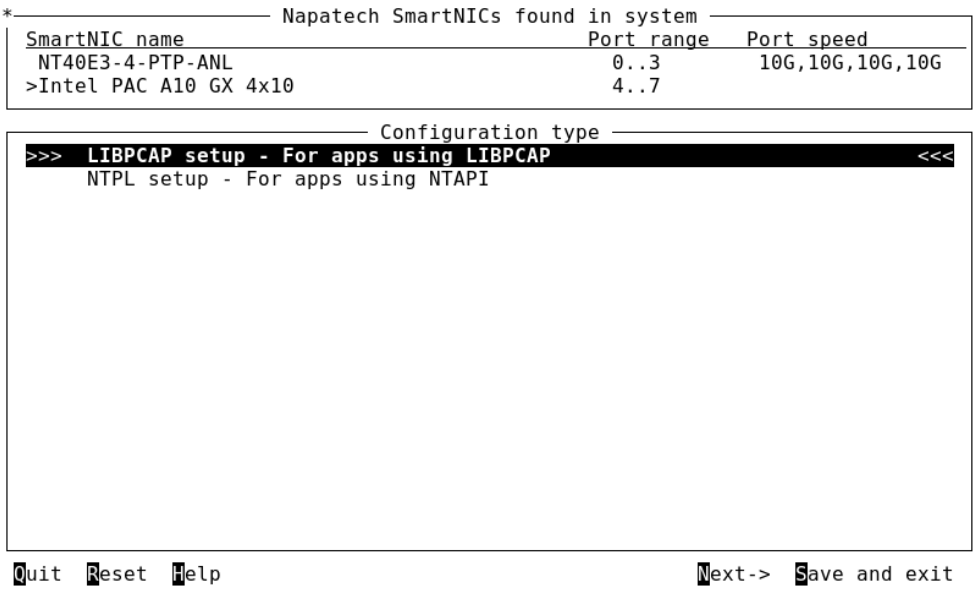
Use the **Configuration type** page to choose which API to configure: libpcap or NTAPI.

The Configuration type page

The upper area displays the available SmartNICs and the port numbering. See SmartNICs and Port Speeds on page 24.

In the middle area, you select one of two configuration types:

- LibPCAP setup - For apps using LIBPCAP
- NTPL setup - For apps using NTAPI



The **setup_wizard** tool stores separate configurations for libpcap and NTAPI in an internal configuration file. To switch between the libpcap and the NTAPI configuration, select the configuration type and press **S** to save and exit.

LIBPCAP setup - For apps using libpcap

For use with applications that support libpcap. When you save this configuration, the driver configuration file `/opt/napatech3/config/ntservice.ini` is modified and the libpcap configuration file `/opt/napatech3/config/ntpcap.ini` is created.

NTPL setup - For apps using NTAPI

For use with applications that support the Napatech API (NTAPI). When you save this configuration, the driver configuration file `/opt/napatech3/config/ntservice.ini` is modified and an NTPL configuration file `/opt/napatech3/config/default.ntpl` is created.

A system key **ntplFileName** in `ntservice.ini`is set to the name of the NTPL configuration file, and the NTPL configuration file runs automatically when the driver is started.

Tip:

Delete the key **ntplFileName** if you do not want the NTPL configuration file to run.

3.1.3 libpcap Interface Configuration

Use the **Interface configuration method** page to set up the mapping between SmartNIC Rx ports and libpcap interfaces.

The Interface Configuration page

The upper area displays the available SmartNICs and the port numbering. See SmartNICs and Port Speeds on page 24.

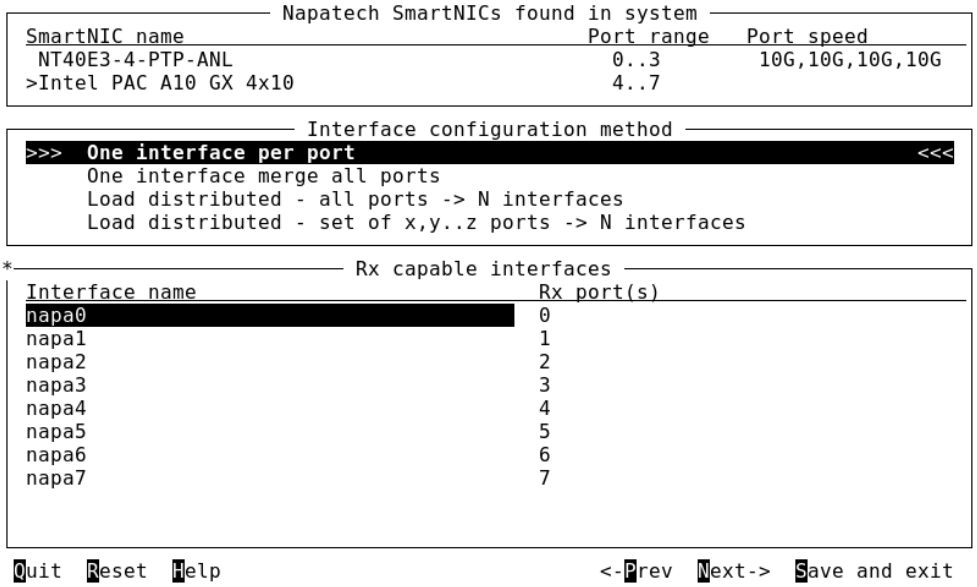
In the middle area, you select one of four interface configuration methods:

- One interface per port
- One interface merge all ports
- Load distributed - all ports -> N interfaces
- Load distributed - set of x,y..z ports -> N interfaces

In the lower area, you can edit interface names and, for **Load distributed - set of x,y..z ports -> N interfaces**, edit which Rx ports map to which libpcap interface.

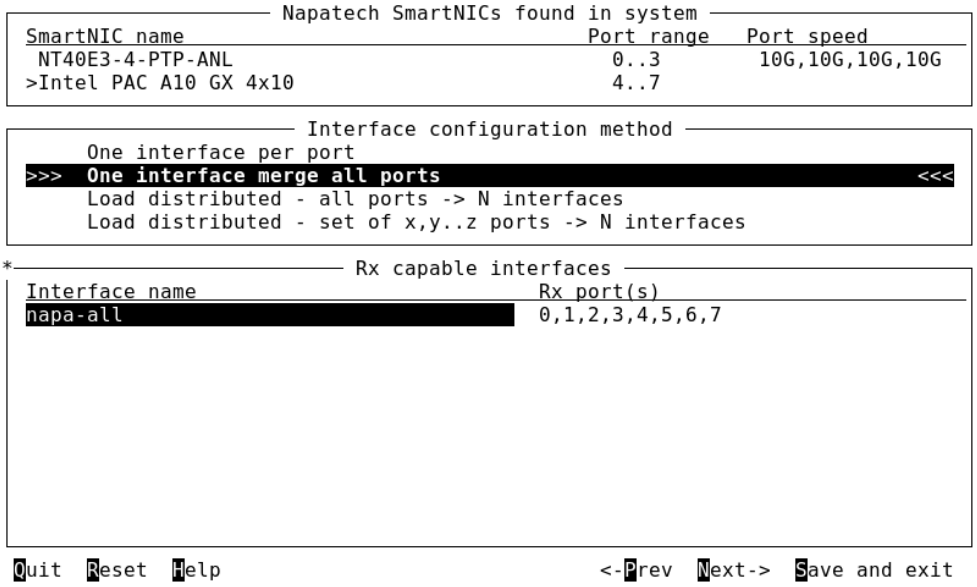
One interface per port

The default configuration is one interface per port. Use this configuration if traffic from each port should be handled by a separate libpcap interface.



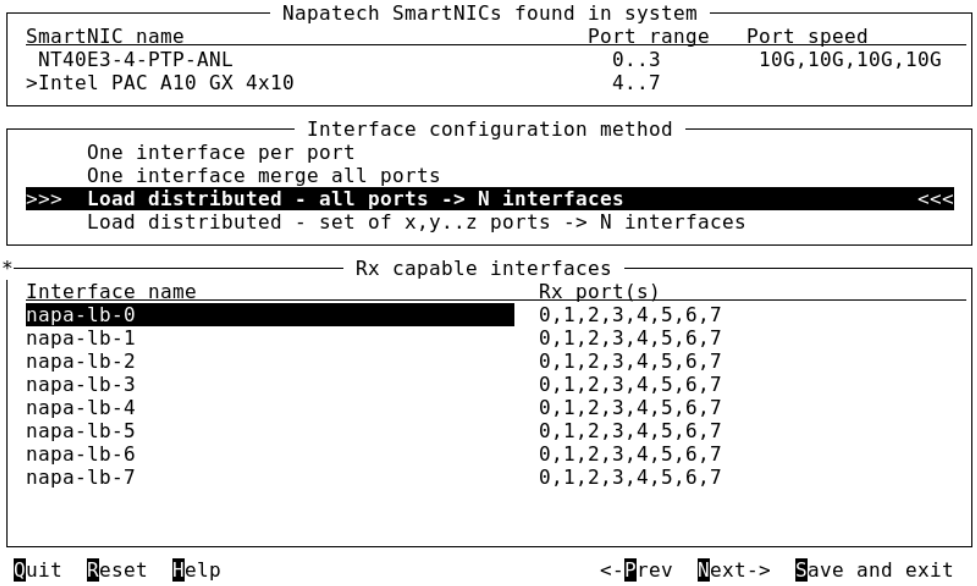
One interface merges all ports

All ports use one interface. Use this configuration to make traffic from all ports available on one libpcap interface, for example, to allow a single process to handle all traffic. You can edit which Rx ports maps to the interface.



Load distributed - all ports -> N interfaces

All ports map to N interfaces to enable load balancing. Use this configuration, for example, to distribute processing of traffic to a number of processes to take advantage of multi-CPU systems.



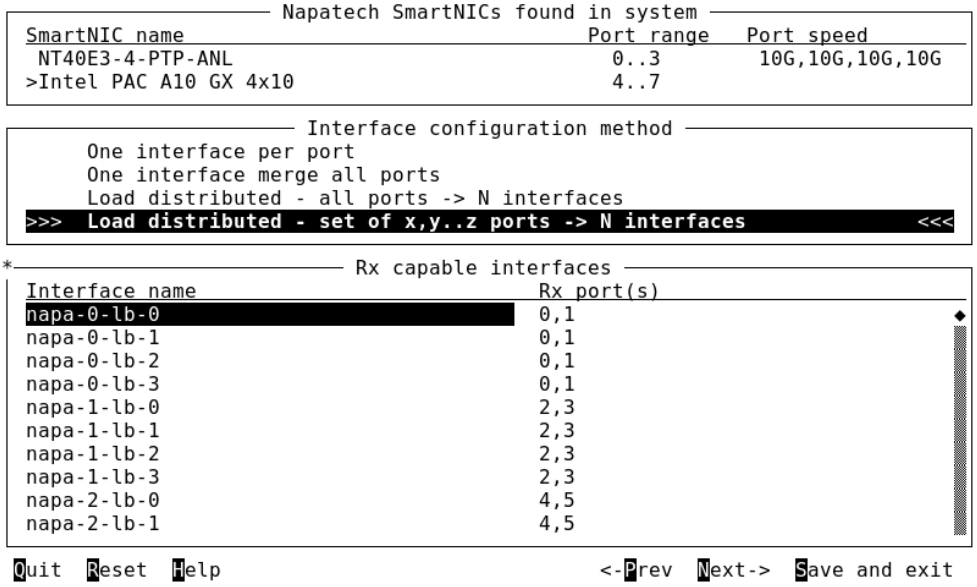
Choose how many libpcap interfaces to create. The default is 8, the range is 2 to 32.

The traffic from all ports is distributed between the defined libpcap interfaces. The way the traffic is distributed depends on the selected hash mode. The hash mode is selected on the **Packet Processing** page.

You can edit which Rx ports map to each libpcap interface.

Load distributed - set of x,y..z ports-> N interfaces

Specific ports map to N interfaces to enable load balancing. Use this configuration to set up any combination of merging and load distribution.



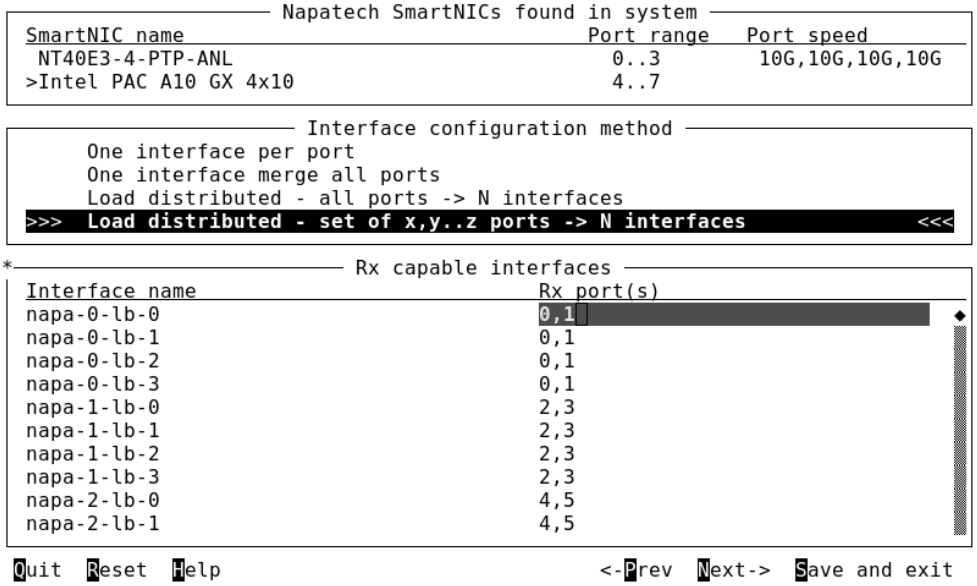
Choose how many libpcap interfaces to create. The default is 8, the range is 2 to 32.

The traffic from the ports is distributed between the defined libpcap interfaces. The way the traffic is distributed depends on the selected hash mode. The hash mode is selected on the **Packet Processing** page.

You can edit which Rx ports map to each libpcap interface.

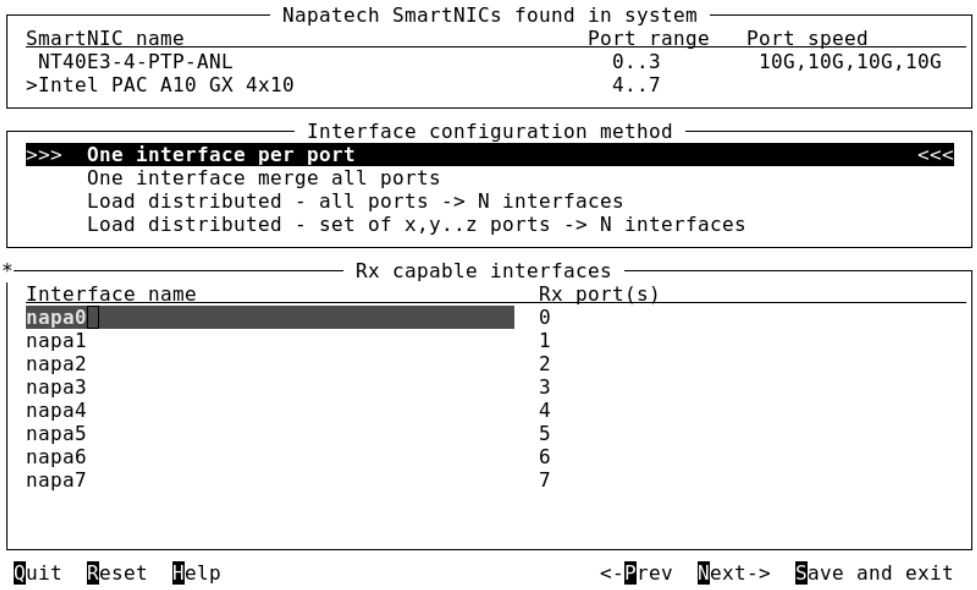
Editing the mapping of ports to interfaces

For **Load distributed - all ports -> N interfaces** and **Load distributed - set of x,y..z ports -> N interfaces**, you can edit which Rx ports map to which libpcap interface.



Editing libpcap interface names

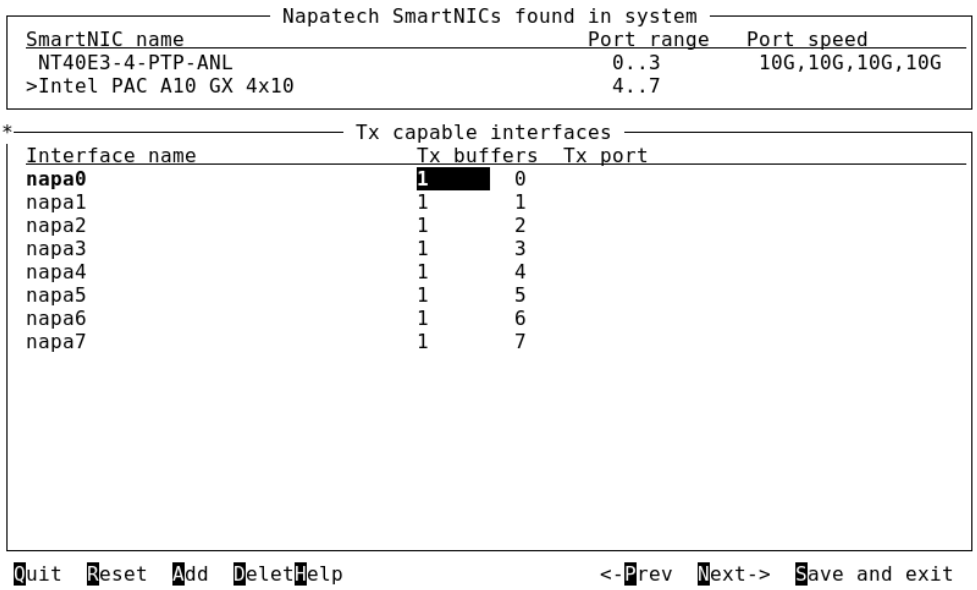
When you have selected an interface configuration method, you can edit the libpcap interface names.



3.1.3.1 Tx Capable Interfaces

Use the **Tx capable interfaces** page to allocate transmit buffers for libpcap interfaces.

The Tx Capable Interfaces page



For each libpcap interface defined on the **Interface configuration method** page, you can allocate buffers and select which SmartNIC port to transmit on:

- For each application or application thread that transmits through a libpcap interface, allocate one Tx buffer.
- Allocate 0 Tx buffers to disable transmit capability on a libpcap interface.

You can add additional transmit-only libpcap interfaces and map them to SmartNIC Tx ports.

Add	Add a transmit-only libpcap interface.
Delete	Delete a transmit-only libpcap interfaces that you have added. You cannot delete the libpcap interfaces defined on the **Interface Configuration** page.

3.1.3.2 Adding an NTPL Expression

You can add an NTPL expression to the configuration in order to take advantage of hardware-accelerated filtering and slicing.

Context

Part of the configuration of Napatech libpcap is stored in the file `/opt/napatech3/config/ntpcap.ini`. To add hardware-accelerated filtering and slicing, you edit NTPL commands in this file.

Note: `ntpcap.ini` is created from scratch, overwriting any manual modifications, each time you exit the **`setup_wizard`** tool using the **Save and exit** command. The previous version of `ntpcap.ini` is renamed `ntpcap.ini.old`.

The ntpcap.ini file

The `ntpcap.ini` file is located in `/opt/napatech3/config/`. This file is part of the configuration created using **`setup_wizard`**.

The `ntpcap.ini` file consists of a number of sections: a `[Common]` section, and one or more subsequent sections that create libpcap interfaces.

The [Common] section

The `[Common]` section contains **`DefineMacro`** commands that make it easier to add filters, and **`Assign`** commands that create *streams* and map SmartNIC ports to these streams.

```
[Common]
Ntpl1 = 'DefineMacro("mMacDestAddr",
"Data[DynOffset=DynOffLayer2Frame;Offset=0;DataType=MacAddr]")'
Ntpl2 = 'DefineMacro("mMacSrcAddr",
"Data[DynOffset=DynOffLayer2Frame;Offset=6;DataType=MacAddr]")'
Ntpl3 = 'DefineMacro("mIPVersion",
"Data[DynOffset=DynOffIPv4Frame;Offset=0;DataType=ByteStr1;DataMask=[7:4]]")'
Ntpl4 = 'DefineMacro("mIPv4SrcAddr",
"Data[DynOffset=DynOffIPv4Frame;Offset=12;DataType=IPv4Addr]")'
Ntpl5 =
'DefineMacro("mIPv4DestAddr","Data[DynOffset=DynOffIPv4Frame;Offset=16;DataType=IPv4Addr]")'
Ntpl6 = 'DefineMacro("mIPv6SrcAddr",
"Data[DynOffset=DynOffIPv6Frame;Offset=8;DataType=IPv6Addr]")'
Ntpl7 =
'DefineMacro("mIPv6DestAddr","Data[DynOffset=DynOffIPv6Frame;Offset=24;DataType=IPv6Addr]")'
Ntpl8 = 'DefineMacro("mUdpSrcPort",
"Data[DynOffset=DynOffUDPFrame;Offset=0;DataType=ByteStr2]")'
Ntpl9 = 'DefineMacro("mUdpDestPort",
"Data[DynOffset=DynOffUDPFrame;Offset=2;DataType=ByteStr2]")'
Ntpl10 = 'DefineMacro("mTcpSrcPort",
"Data[DynOffset=DynOffTCPFrame;Offset=0;DataType=ByteStr2]")'
Ntpl11 = 'DefineMacro("mTcpDestPort",
"Data[DynOffset=DynOffTCPFrame;Offset=2;DataType=ByteStr2]")'
Ntpl12 = "Assign[StreamId = 0] = Port == 0"
Ntpl13 = "Assign[StreamId = 1] = Port == 1"
Ntpl14 = "Assign[StreamId = 2] = Port == 2"
Ntpl15 = ...
...

[napa0]
StreamId = 0
Tx = 0

[napa1]
StreamId = 1
Tx = 1
```

To add filtering and slicing, you edit the **`Assign`** commands in the `[Common]` section. Each NTPL command in the `[Common]` section is prepended with a numbered key, `Ntpl1`, `Ntpl2`, `Ntpl3`, etc.

Note: The `Ntpl1`, `Ntpl2`, etc. keys are not shown in the examples. If you need to add a new **`Assign`** command, prepend it with a numbered `Ntpl`*`nn`* key and renumber subsequent `Ntpl`*`nn`* keys.

For filtering, add conditions in the expression on the right side of the **`Assign`** command.

For slicing, add options in the square brackets on the left side of the **`Assign`** command.

libpcap interface sections

The streams created in the `[Common]` section are mapped to libpcap interface names in the subsequent sections.

You do not need to edit these sections.

Note:

You can map a stream to more than one libpcap interface in order to share resources (host buffers). In this case, the slowest application that open one of these libpcap interface blocks the other interfaces mapped to this stream.

Step	Action
1	Use the interactive **`setup_wizard`** tool to create the initial configuration.
2	Open `ntpcap.ini` in a text editor, and locate the **`Assign`** command that create the `StreamId` used by the libpcap interface you want to accelerate.
3	Add filter expressions and **`Assign`** options.
4	Save `ntpcap.ini`.

Result

`ntpcap.ini` take effect as soon as all currently open Napatech libpcap interface are closed and a Napatech libpcap interface is then opened.

Note:

You do not have to restart **`ntservice`**.

3.1.3.3 NTPL Filter Examples

Use these examples to add hardware-accelerated filtering on ports, IP addresses, protocols and errored packets.

Adding NTPL filter expressions

Refer to Adding an NTPL Expression on page 31 for instructions on how to add NTPL expressions to the configuration.

Filter on ports

Filter on a port, use mTcpDestPort for TCP destination port and mTcpSrcPort for TCP source port:

```
Assign[StreamId=0] = (Port == 0) AND (mTcpDestPort==8080)
```

Filter on several ports:

```
Assign[StreamId=0] = (Port == 0) AND (mTcpDestPort == 8080, 8081,
8089)
```

Filter on a range of ports:

```
Assign[StreamId=0] = (Port == 0) AND (mTcpSrcPort == (0x100..0x400))
```

Filtering on UDP ports can be done in the same way.

Filter on IP addresses

Filter on an IP address, mIPv4SrcAddr for IPv4 source address, and mIPv4DestAddr for IPv4 destination address:

```
Assign[StreamId=0] = (Port == 0) AND (mIPv4SrcAddr == [192.168.1.1])
```

Filter on an IP range of addresses (192.168.1.10 – 192.168.1.20):

```
Assign[StreamId=0] = (Port == 0) AND (mIPv4SrcAddr ==
([192.168.1.10]..[192.168.1.20]))
```

Filter on an IP subnet (all addresses from 192.168.1.0 to 192.168.1.255 using a bit mask):

```
Assign[StreamId=0] = (Port == 0) AND (mIPv4SrcAddr ==
{[FF.FF.FF.00]:[192.168.1.0]})
```

Filter on protocols

Filter on a protocol:

```
Assign[StreamId=0] = (Port == 0) AND (layer3protocol == IPv4)
```

Filter on several protocols:

```
Assign[StreamId=0] = (Port == 0 ) AND (Layer4Protocol == UDP, TCP,
ICMP)
```

Filter on errored packets

Look for errored packets:

```
Assign[StreamId=0] = (CrcError == TRUE OR (IpChecksumError == TRUE
OR (UdpChecksumError == TRUE OR TcpChecksumError == TRUE)))
```

3.1.3.4 NTPL Slicing Examples

Adding NTPL slicing expressions

Refer to Adding an NTPL Expression on page 31 for instructions on how to add NTPL expressions to the configuration.

Slicing

This NTPL example captures TCP frames from Port 0 and slices them to 64 bytes plus the size of any layer 2, 3 and 4 protocol headers:

```
Assign[StreamId = 0; SliceOffset = 64; SliceAddHeader = Layer2And3And4HeaderSize] = (Port == 0) AND (Layer4Protocol
 == TCP)
```

For more information, refer to DN-1128 and DN-0449.

3.1.4 NTPL Interface Configuration

Use the **Stream configuration method** page to set up the mapping between SmartNIC Rx ports and NTAPI streams.

The Stream Configuration page

The upper area displays the available SmartNICs and the port numbering. See SmartNICs and Port Speeds on page 24.

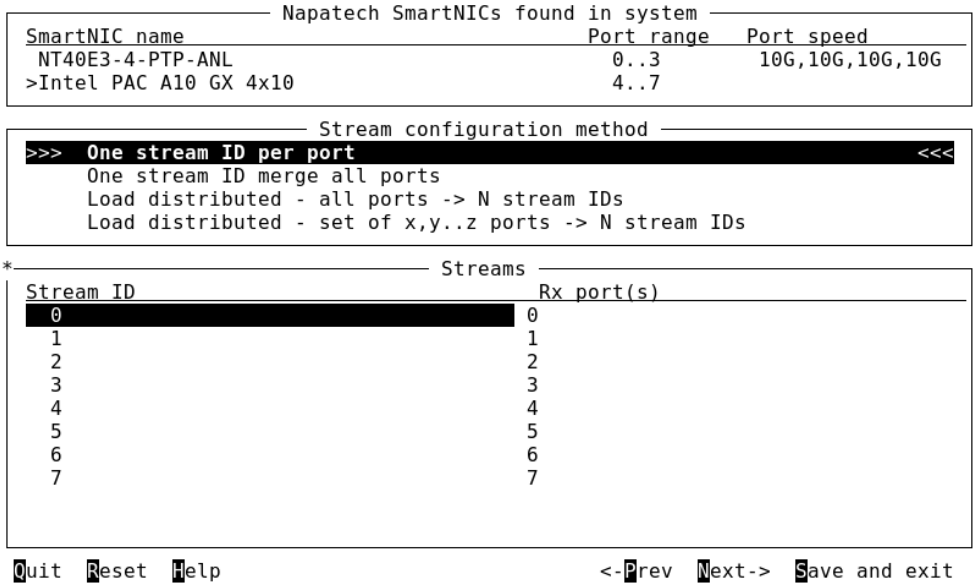
In the middle area, you select one of four interface configuration methods:

- One stream ID per port
- One stream ID merge all ports
- Load distributed - all ports -> N stream IDs
- Load distributed - set of x,y..z ports -> N stream IDs

In the lower area, you can edit stream IDs and, for **Load distributed - set of x,y..z ports -> N stream IDs**, edit which Rx ports maps to which stream ID.

One stream ID per port

The default configuration is one port per stream. Use this configuration if traffic from each port should be handled by a separate thread or application.

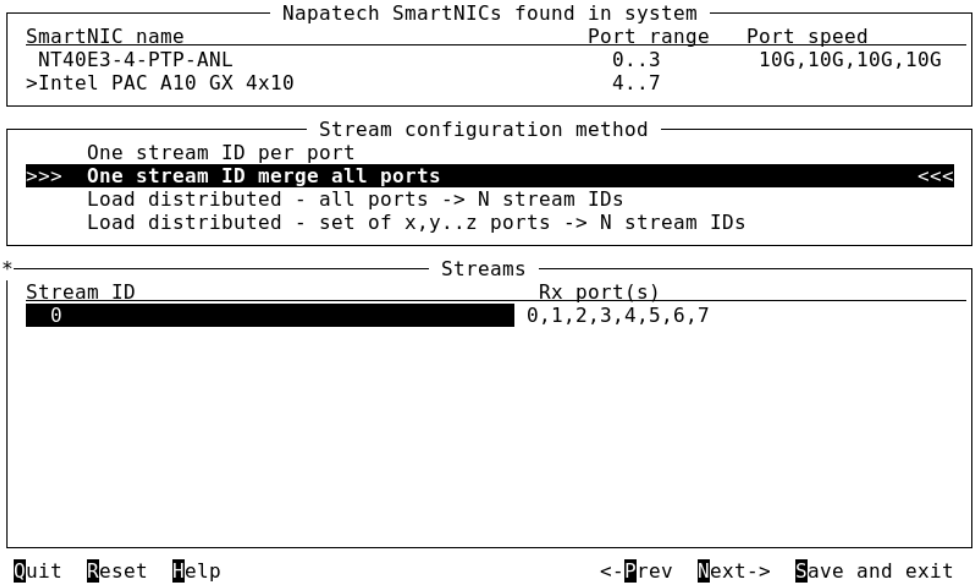


The NTPL generated by this configuration example is as follows:

```
Assign[StreamId = 0] = Port == 0
Assign[StreamId = 1] = Port == 1
Assign[StreamId = 2] = Port == 2
Assign[StreamId = 3] = Port == 3
Assign[StreamId = 4] = Port == 4
Assign[StreamId = 5] = Port == 5
Assign[StreamId = 6] = Port == 6
Assign[StreamId = 7] = Port == 7
```

One stream ID merge all ports

All ports use one stream. Use this configuration to make traffic from all ports available on one stream, for example, to allow a single process to handle all traffic.

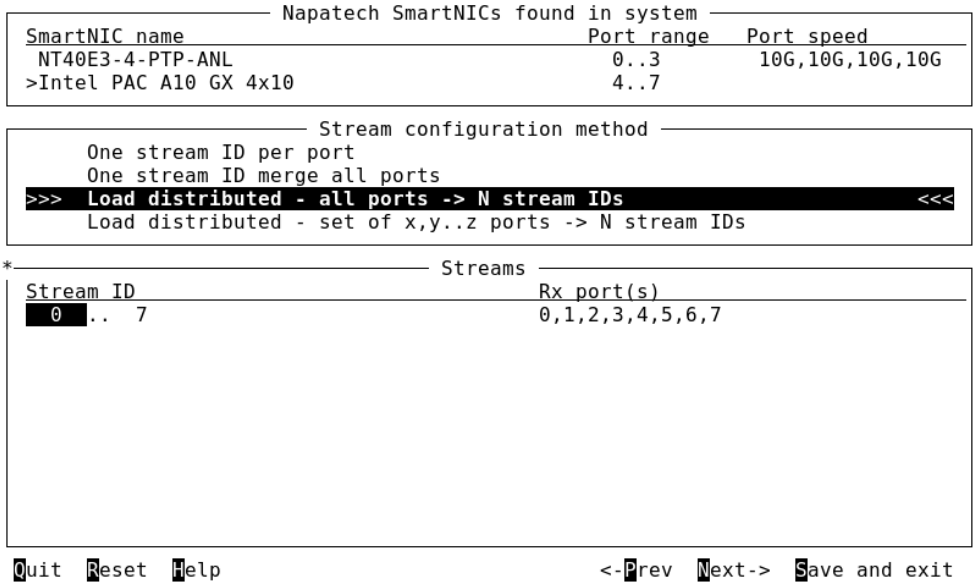


The NTPL generated by this configuration is as follows:

```
Assign[StreamId = 0] = Port == 0,1,2,3,4,5,6,7
```

Load distributed - all ports -> N stream IDs

All ports map to N streams to enable load balancing. Use this configuration, for example, to distribute processing of traffic to a number of processes to take advantage of multi-CPU systems.



The NTPL generated by this configuration is as follows:

```
Assign[StreamId = 0] = Port == 0,1,2,3,4,5,6,7
HashMode = HashRoundRobin
```

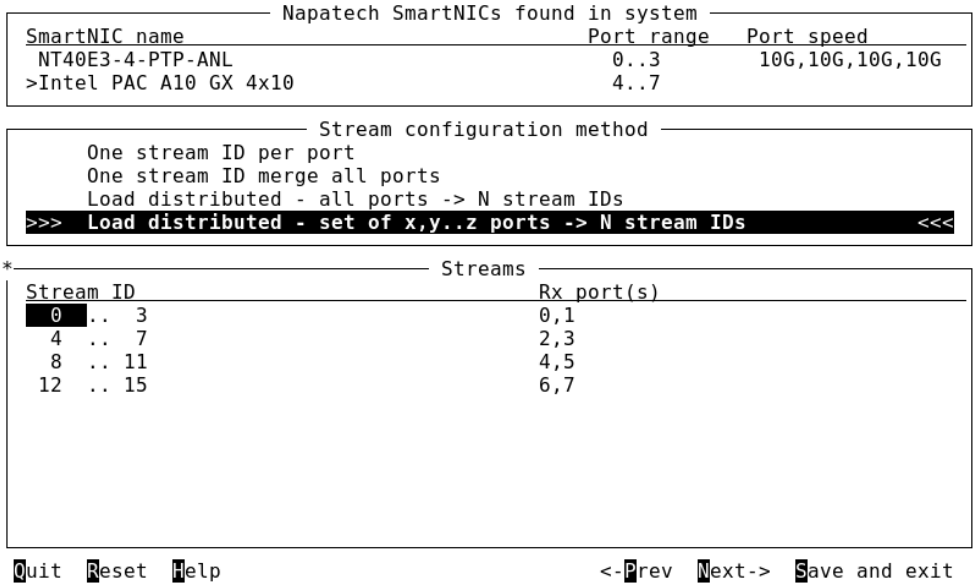
Choose how many streams to create. The default is 8, the range is 1 to 64 for Intel® PAC A10 GX acceleration cards, and 1 to 128 for Napatech SmartNICs.

The traffic from all ports is distributed between the defined streams. The way the traffic is distributed depends on the selected hash mode. The hash mode is selected on the**Packet Processing** page.

You can edit which Rx ports map to each stream ID.

Load distributed - set of x,y..z ports-> N stream IDs

Specific ports map to N streams to enable load balancing. Use this configuration to setup any combination of merging and load distribution.



The NTPL generated by this configuration is as follows:

```
Assign[StreamId = (0..3)] = Port == 0, 1
Assign[StreamId = (4..7)] = Port == 2, 3
Assign[StreamId = (8..11)] = Port == 4, 5
Assign[StreamId = (12..15)] = Port == 6, 7
HashMode = HashRoundRobin
```

Choose how many streams to create. The range is 1 to 64 for Intel® PAC A10 GX acceleration cards, and 1 to 128 for Napatech SmartNICs.

The traffic from all ports is distributed between the defined streams. The way the traffic is distributed depends on the selected hash mode. The hash mode is selected on the **Packet Processing** page.

You can edit which Rx ports map to each stream ID.

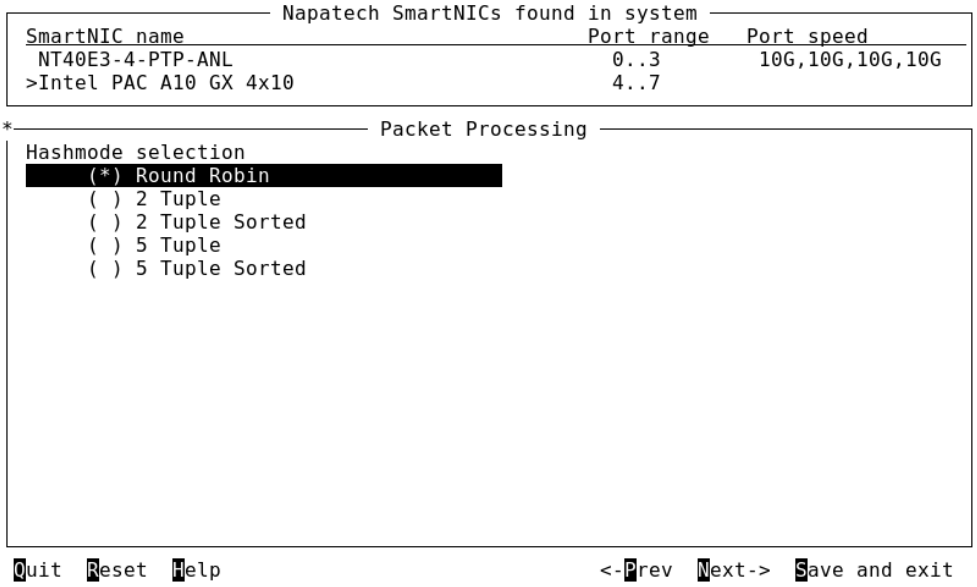
3.1.5 Packet Processing

Use the **Packet Processing** page to select an algorithm for load distribution.

The Packet Processing page

The upper area displays the available SmartNICs and the port numbering. See SmartNICs and Port Speeds on page 24.

In the lower area, you can select the algorithm that is be used for load distribution if you have selected a load-distributing libpcap interface or NTAPI stream configuration.



Hash mode	Description
Round Robin	Packets are distributed using a round-robin method.
2 Tuple	The hash key is calculated from the source and destination IP addresses as they appear in the header.
2 Tuple Sorted	Similar to 2 Tuple, but sorted before the hash key calculation so that all traffic between two endpoints is distributed to the same libpcap interface.
5 Tuple	The hash key is calculated from the source and destination IP addresses as they appear in the header, the source and destination ports, and the protocol.
5 Tuple Sorted	Similar to 5 Tuple, but sorted before the hash key calculation so that all traffic between two endpoints is distributed to the same libpcap interface.

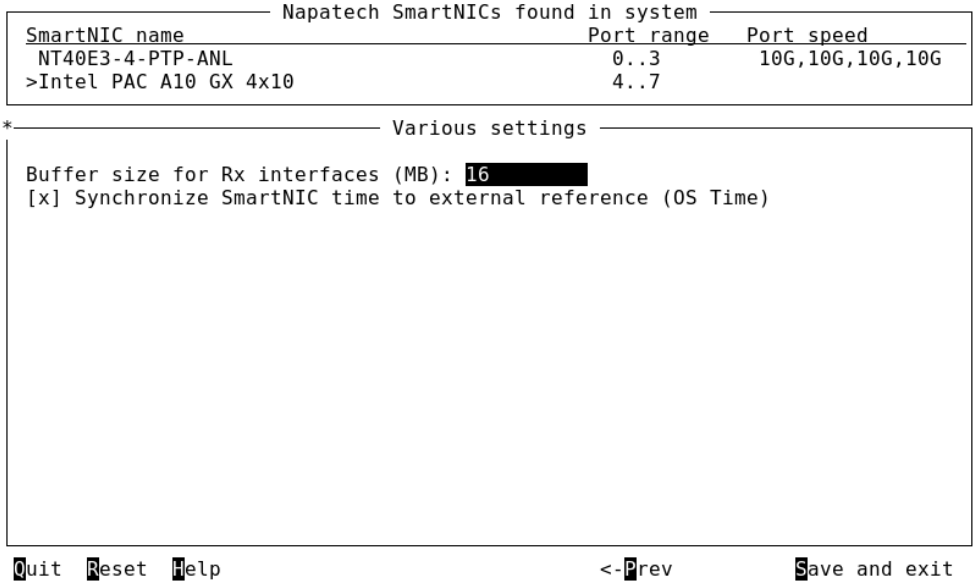
3.1.6 Various Settings

Use the **Various Settings** page to adjust the Rx interface buffer size and to control time synchronization.

The Various Settings page

The upper area displays the available SmartNICs and the port numbering. See SmartNICs and Port Speeds on page 24.

In the lower area, you can adjust Rx interface buffer size and time synchronization.



You can adjust the buffer size for Rx interfaces in increments of 4 MB. The minimum size is 16 MB.

By default, time-stamping on all SmartNICs is based on the internal clock reference on each SmartNIC. To ensure synchronous time-stamping across the SmartNICs, you can set the internal clock reference on each SmartNIC to be synchronized with the OS time on the server.

4 Applications

Installation and configuration guides for using some popular open-source security applications with Napatech Link™ Capture Software.

Some popular security applications

This guide includes installation and configuration guides for the following open source security applications:

- Wireshark, Tcpdump and other libpcap-based applications
- DPDK-based applications
- Suricata
- Snort
- Bro
- TRex

Installation and API options

Depending on the application, you can choose between several APIs. If package-based installation is a priority, libpcap may be the best option. If you are willing to build the application from source, you may get better performance and more features.

Application	Napatech libpcap	Native NTAPI	Napatech DPDK
Wireshark, Tcpdump, Tcpreplay, etc.	In general, packages. Some applications might need to be built from source.		
Suricata	Suricata packages	Build Suricata from source.	
Snort	Packages for Snort 2.9.x and DAQ 2.0.x		Build Snort 3.0 and DAQ 2.2.2 from source. Requires Napatech DPDK installed.
Bro	Bro packages	Build Bro from source.	
TRex			Build from source. DPDK libs and drivers are included in TRex source.

Pros and cons of interface options

When you decide which API to use, consider these advantages and disadvantages:

	Napatech libpcap	Native NTAPI	Napatech DPDK
Advantages	libpcap support is available for many applications. libpcap provides simple package-based installation.	Native interfaces to NTAPI can provide multi-threaded and otherwise optimized packet processing.	DPDK provides multi-threaded/ multi-stream and otherwise optimized packet processing.
Disadvantages	libpcap runs single-threaded and requires more work to optimize utilization of multiple CPUs.	Native interfaces to NTAPI are not available for many applications, and usually require building from source.	DPDK is not available for many applications, and usually requires building from source.

This chapter contains the following sections:

4.1 Installing and Running libpcap Applications

You can install most libpcap-based applications for use with Napatech libpcap interfaces using your distributions package manager.

Prerequisite

If the user group `napatech` exists, you must have root privileges or be a member of the `napatech` group in order to use Napatech libpcap interfaces.

Dynamically linked applications should work the same whether installed before or after the Napatech libpcap.

Applications that are statically linked to libpcap must be recompiled and linked to the Napatech libpcap.

Ensure the use of the Napatech libpcap library

If other libpcap libraries are installed on the target platform, it is uncertain which libpcap library is used. There are two ways to ensure that the Napatech libpcap library is used:

- By removing all other libpcap libraries from the system
- By forcing the application to use the Napatech libpcap library

To force an application to use the Napatech libpcap library, use one of these methods:

Set **`LD_PRELOAD`** in the environment for the application:

```
LD_PRELOAD=/opt/napatech3/lib/libpcap.so <app>
```

Set **`LD_LIBRARY_PATH`** in the environment for the application:

```
LD_LIBRARY_PATH=/opt/napatech3/lib/ <app>
```

Common applications are available from your distribution

Most common libpcap-based applications can be installed from your distributions default package repositories, and run without further configuration. The package name depends on the distribution.

For example, to install and run Wireshark on RHEL/CentOS:

```
sudo yum install wireshark-gnome
wireshark
```

On Ubuntu and derivatives of Ubuntu, for example, Mint, you must preload the Napatech libpcap shared library like this:

```
sudo apt-get install wireshark
LD_PRELOAD=/opt/napatech3/lib/libpcap.so wireshark
```

Extra repositories for popular applications

If a package is not available from your distributions default package repositories, add a repository that offers the package. The application homepage might offer detailed instructions about how to add the repository and install the package.

For example, for Bro, the following is suggested for CentOS 7 at https://www.bro.org/download/packages.html:

```
cd /etc/yum.repos.d/
sudo wget http://download.opensuse.org/repositories/network:bro/CentOS_7/network:bro.repo
sudo yum install bro
```

Package downloads for popular applications

For other applications you might have to download the package and install it locally. Snort is an example of this:

- Packages for the latest release of Snort and daq are available at https://www.snort.org/downloads/.
- Documentation and setup guides for various distributions are available at https://www.snort.org/documents.
- You can get earlier source releases from http://sourceforge.net/projects/snort/.

For more information about installing and running Snort with SmartNICs and Napatech Software Suite, see Snort on page 52 and DN-0666.

Application must be recompiled from source

If you need to compile the application yourself from source because no package is available for your distribution, the process depends on the configuration options of the application.

Note:

You should preferably link to the Napatech libpcap library to ensure the best match to the version of the Napatech libpcap.

The following examples use the installation directory prefix (`--prefix`) `/opt/napatech3`.

Most applications rely on Autoconf for configuration, and configure script options allow you to select Napatech libpcap headers and libraries. You must usually specify linker flags.

libpcap configure script option

For example, for Tcpreplay there is a `--with-libpcap` option that helps the configure script find Napatech libpcap:

```
wget https://github.com/appneta/tcpreplay/releases/download/v4.1.0/tcpreplay-4.1.0.tar.gz
tar xzf tcpreplay-4.1.0.tar.gz
cd tcpreplay-4.1.0
./configure --prefix=/opt/napatech3 LDFLAGS="-L/opt/napatech3/lib -lntapi -lntos"
--with-libpcap=/opt/napatech3 --enable-dynamic-link --enable-force-inject
make
sudo make install
```

Even if a `--with-libpcap` option is available, you might also need to set compiler flags for the configure script to find Napatech libpcap. This is the case for recent versions of Tcpreplay:

```
wget https://github.com/appneta/tcpreplay/releases/download/v4.2.4/tcpreplay-4.2.4.tar.gz
cd tcpreplay-4.2.4
./configure --prefix=/opt/napatech3 CFLAGS=-I/opt/napatech3/include
LDFLAGS="-L/opt/napatech3/lib -lntapi -lntos" --with-libpcap=/opt/napatech3
--enable-dynamic-link --enable-force-inject
...
```

Compiler and linker flags

For Tcpdump, you have to specify the Napatech libpcap using compiler and linker flags:

```
wget http://www.tcpdump.org/release/tcpdump-4.9.0.tar.gz
tar xzf tcpdump-4.9.0.tar.gz
cd tcpdump-4.9.0
./configure --prefix=/opt/napatech3 CFLAGS=-I/opt/napatech3/include
LDFLAGS="-L/opt/napatech3/lib -lntapi -lntos"
make
sudo make install
```

Some configuration scripts do not allow compiler and linker flags as options. In that case, set the flags in the environment:

```
export CFLAGS=-I/opt/napatech3/include
export LDFLAGS="-L/opt/napatech3/lib -lntapi -lntos"
./configure --prefix=/opt/napatech3 ...
...
```

4.2 Installing Napatech DPDK

Napatech DPDK is a fork of DPDK that supports Napatech drivers and SmartNICs.

DPDK documentation

Installing, compiling and running DPDK is quite complex. A full description can be found here:

- http://dpdk.org/doc/quick-start
- http://dpdk.org/doc/guides/linux_gsg/index.html.

Napatech DPDK readme

The latest Napatech DPDK `readme` file can be found at https://github.com/napatech/dpdk/blob/master/ntacc_readme.md

The `readme` file contains information on how to compile Napatech DPDK and set up the Napatech driver for DPDK.

Getting and compiling Napatech DPDK

Napatech DPDK can be downloaded or cloned from: https://github.com/napatech/dpdk/releases.

The environment variable **`NAPATECH3_PATH`** must be set to the Napatech driver installation root before compiling DPDK. The default installation root is `/opt/napatech3`.

```
# export NAPATECH3_PATH=/opt/napatech3
```

To get and compile Napatech DPDK, locate the latest release at https://github.com/napatech/dpdk/releases. In this example, the `v18.08.0_1.9` release is used:

```
# wget https://github.com/napatech/dpdk/archive/v18.08.0_1.9.tar.gz
# tar xvf v18.08.0_1.9.tar.gz
# cd dpdk-v18.08.0_1.9/
# make config T=x86_64-native-linuxapp-gcc install
# make -j
```

DPDK and hugepages

To run any DPDK application, the Linux kernel option `hugepages` must be enabled.

Edit the file `/etc/default/grub` and change or add to the following line:

```
GRUB_CMDLINE_LINUX_DEFAULT="default_hugepagesz=1G hugepagesz=1G
hugepages=2"
```

If you are using UEFI based boot, run this after editing `/etc/default/grub`:

```
sudo grub2-mkconfig -o /boot/efi/EFI/fedora/grub.cfg
```

Else run this:

```
sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

4.3 Suricata

Installing and configuring Suricata for use with Napatech Link™ Capture Software.

Introduction

Suricata is an open source intrusion detection and prevention engine.

Suricata can be installed for use with Napatech Link™ Capture Software in two ways:

- To get optimal performance, build Suricata from source with native Napatech support enabled.
- Suricata can be installed from binary packages and use libpcap interfaces to Napatech Link™ Capture Software.

  Note: With this option, there is no support for Napatech hardware.

Package-based installation

Identify a suitable precompiled Suricata package for your OS distribution. See Installing and Running libpcap Applications on page 42 for information about how to install and use package-based applications with Napatech libpcap.

See DN-0428 for more information about configuration of Napatech libpcap.

For Ubuntu, the OISF maintains a PPA, `suricata-stable`, that always contains the latest stable release:

```
sudo add-apt-repository ppa:oisf/suricata-stable
sudo apt-get update
sudo apt-get install suricata
```

For RedHat Enterprise Linux 7 and CentOS 7 the EPEL repository can be used:

```
sudo yum install epel-release
sudo yum install suricata
```

See also http://suricata.readthedocs.io/en/latest/install.html#install-binary-packages.

Dependencies

You need to install the following libraries and their development headers:

- libpcap
- libpcre
- libmagic
- zlib
- libyaml

You need the following tools:

- make
- gcc
- pkg-config

Building Suricata from source

To build and run Suricata with Napatech support:

1. Download and extract the Suricata source: https://github.com/OISF/suricata.
2. Run **`configure`** to enable Napatech support and prepare for compilation:

```
./configure --enable-napatech
--with-napatech-includes=/opt/napatech3/include
--with-napatech-libraries=/opt/napatech3/lib
make
sudo make install-full
```

3. Edit the `suricata.yaml` file to configure the maximum number of streams to use:

   If you plan to use the load distribution (RSS-like) feature in the SmartNIC, the list should contain the same number of streams as host buffers defined in `ntservice.ini`:

```
Napatech :
    # The Host Buffer Allowance for all streams
    # (-1 = OFF, 1 - 100 = percentage of the host buffer that can be held back)
    hba : - 1

    # use_all_streams set to "yes" queries the Napatech service for all configured
    # streams and listen on all of them. When set to "no" the streams config array
    # is used.
    use - all - streams : yes

    # The streams to listen on
    streams : [ 0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 ]
```

   Note: `hba` is useful only when a stream is shared with another application. When `hba` is enabled, packets are dropped (i.e. not delivered to Suricata) when the host-buffer utilization reaches the high-water mark indicated by the `hba` value. This ensures that, even if Suricata is slow with packet processing, the other application still receives all of the packets. If `hba` is enabled without another application sharing the stream, it results in sub-optimal packet buffering.

   When setting `use-all-streams=yes`, Suricata uses all available streams created by the driver ([Advanced multi-threaded configuration]). When setting `use-all-streams=no`, Suricata uses the streams listed in streams: [0, 1, 2, 3, 4, 5, 6, 7], in this case the streams from 0 to 7. The number is the stream ID used in the NTPL command.

4. Configure Suricata:

   For the basic installation, set up the SmartNIC to merge all physical ports into a single stream that Suricata can read from. For this configuration, Suricata handles the packet distribution to multiple threads.

   Change these lines in `/opt/napatech3/bin/ntservice.ini` for best single buffer performance:

```
TimeSyncReferencePriority = OSTime      # Timestamp clock synchronized to the OS
HostBuffersRx = [ 1 , 16 , -1 ]         # [number of host buffers, Size(MB), NUMA node]
```

5. Stop and restart **`ntservice`** after making changes to `ntservice.ini`:

```
sudo /opt/napatech3/bin/ntstop.sh
sudo /opt/napatech3/bin/ntstart.sh
```

6. Test your setup:

   Create a file with the following commands:

```
Delete=all                                  # Delete any existing filters
Assign[priority=0; streamid=0]= all         # Assign all physical ports to stream ID 0
```

Run the commands using the **ntpl** tool:

```
sudo /opt/napatech3/bin/ntpl -f <my_ntpl_file>
```

7. Start Suricata:

```
sudo suricata -c /usr/local/etc/suricata/suricata.yaml --napatech --runmode workers
```

Advanced multi-threaded configuration

A more advanced configuration uses the load distribution (RSS - like) capability in the SmartNIC. Create 8 streams and set up the SmartNIC to distribute the load based on a 5-tuple hash. Increasing buffer size minimizes packet loss only if your CPU cores are fully saturated. Setting the minimum buffer size (16MB) gives the best performance (minimize L3 cache hits) if your CPU cores have capacity.

1. Modify the `ntservice.ini` file to increase the number and size of the host buffers:

```
TimeSyncReferencePriority = OSTime    # Timestamp clock synchronized to the OS
HostBuffersRx = [8,16,-1]             # [number of host buffers, Size (MB), NUMA node]
```

Note: Setting the NUMA node parameter to -1 causes the driver to allocate from the same NUMA node where the SmartNIC is located.

2. Stop and restart **ntservice** after making changes to `ntservice.ini`:

```
sudo /opt/napatech3/bin/ntstop.sh
sudo /opt/napatech3/bin/ntstart.sh
```

3. Assign the streams to host buffers and configure the load distribution.

   The load distribution is set up to support both tunneled and non-tunneled traffic. Create a file that contains these NTPL commands:

```
Delete=All                              # Delete any existing filters
HashMode[priority=4]=Hash5TupleSorted
Assign[priority=0; streamid=(0..7)]= all
```

4. Run the NTPL commands using the **ntpl** tool:

```
sudo /opt/napatech3/bin/ntpl -f <my_ntpl_file>
```

5. Run Suricata:

```
sudo suricata -c /usr/local/etc/suricata/suricata.yaml --napatech --runmode workers
```

Manually configuring NUMA nodes

Important: The NUMA node the host buffers must be defined on is the same physical CPU socket that the SmartNIC is plugged into.

The NUMA node for the SmartNIC is entered in `ntservice.ini`:

```
TimeSyncReferencePriority = OSTime      # Timestamp clock synchronized to the OS
HostBuffersRx = [8,16,0]                # [number of host buffers, Size (MB), NUMA node]
```

Each stream ID must be tied to a NUMA node using the NTPL **`Setup`** command:

```
Delete=All                          # Delete any existing filters
Setup[numaNode=0] = streamid==0
Setup[numaNode=0] = streamid==1
Setup[numaNode=0] = streamid==2
Setup[numaNode=0] = streamid==3
Setup[numaNode=0] = streamid==4
Setup[numaNode=0] = streamid==5
Setup[numaNode=0] = streamid==6
Setup[numaNode=0] = streamid==7
HashMode[priority=4]=Hash5TupleSorted
Assign[priority=0; streamid=(0..7)]= all
```

Host buffers versus stream IDs

The number of host buffers must match the number of stream IDs used. In this case, stream ID 0 to 7 are used:

```
Assign[priority=0; streamid=(0..7)]= all
```

The smallest number of host buffers allocated in `ntservice.ini` must be 8:

```
HostBuffersRx = [8,16,-1]
```

Counters

For each stream that is processed the following counters are output in `stats.log`:

- nt<streamid>.pkts - The number of packets received by the stream.
- nt<streamid>.bytes - The total number of bytes received by the stream.
- nt<streamid>.drop - The number of packets that were dropped from this stream due to buffer overflow conditions.

If `hba` is enabled, the following counter is also provided:

- nt<streamid>.hba_drop - the number of packets dropped because the host buffer allowance high-water mark was reached.

In addition to counters host buffer utilization is tracked and logged. This is also useful for debugging. Log messages are output for both host buffers and onboard buffers when 25, 50, and 75 percent of utilization is reached. Corresponding messages are output when utilization decreases.

Related links

Suricata site: https://suricata-ids.org/

Suricata source: https://github.com/OISF/suricata

Suricata documentation: https://suricata-ids.org/docs/

Suricata installation from source: http://suricata.readthedocs.io/en/latest/install.html#source

Suricata binary packages:
http://suricata.readthedocs.io/en/latest/install.html#install-binary-packages

Installation guide for Napatech support:
http://suricata.readthedocs.io/en/latest/capture-hardware/napatech.html

Video guide for Suricata deployment with Napatech:
https://films.napatech.com/suricata-ids-deployment-with-napatech

4.4 Snort

Installing and configuring Snort for Napatech Link™ Capture Software.

Introduction

Snort is an open source network intrusion prevention and detection system.

Snort can be installed for use with Napatech Link™ Capture Software in two ways:

- Snort can be installed from binary packages and use libpcap interfaces to Napatech Link™ Capture Software.
- To get optimal performance and inline support, build Snort 3.0 and DAQ 2.2.2 with DPDK support from source.

Package-based installation

Identify a suitable precompiled Snort package for your OS distribution. See Installing and Running libpcap Applications on page 42 for general information about how to install and use package-based applications with Napatech libpcap.

See DN-0428 for more information about configuration of Napatech libpcap.

See DN-0666 for more information about optimizing Snort with Napatech libpcap.

Installing and running Snort with Napatech DPDK

The Napatech driver must be installed and running before Snort with Napatech DPDK can be used. See Installing Napatech Link™ Capture Software on page 13.

Napatech DPDK must be installed and built before Snort with Napatech DPDK can be used. See Installing Napatech DPDK on page 44.

Get and Compile DAQ for DPDK support

```
# wget
https://github.com/napatech/daq_dpdk_multiqueue/archive/dpdk18.05_mq.zip
# unzip unzip dpdk18.05_mq.zip
# cd daq_dpdk_multiqueue-dpdk18.05_mq/daq-2.2.2/
# export RTE_SDK=/project/dpdk-master/
# ./configure --prefix=/opt/snort
# make
# make install
```

Get and compile Snort 3.0 using DAQ 2.2.2

```
# wget
https://www.snort.org/downloads/snortplus/snort-3.0.0-243-cmake.tar.gz
# tar xvf tar xvf snort-3.0.0-243-cmake.tar.gz
# export PATH=/opt/snort/bin/:$PATH
# ./configure_cmake.sh --prefix=/opt/snort
--with-daq-libraries=/opt/snort/lib
--with-daq-includes=/opt/snort/include
# cd build
# make
# make install
```

Running Snort

Snort in inline mode (IPS) using 14 RSS queues and 14 cores on 1 interface pair:

```
# taskset -c 0-13 /opt/snort/bin/snort --daq dpdk --daq-var
dpdk_argc="-n4" -i "dpdk0:dpdk1" -Q -z 14
```

Snort in inline mode (IPS) using 2 RSS queues and 14 cores on 2 interface pairs:

```
# taskset -c 0-13 /opt/snort/bin/snort --daq dpdk --daq-var
dpdk_argc="-n4" -i "dpdk0:dpdk1 dpdk2:dpdk3" -Q -z 14
```

Snort in passive mode (IDS) using 14 RSS queues and 14 cores on 2 interfaces:

```
# taskset -c 0-13 /opt/snort/bin/snort --daq dpdk --daq-var
dpdk_argc="-n4" -i "dpdk0 dpdk1" -z 14
```

Snort in passive mode (IDS) using 2 RSS queues and 4 cores on 1 interfaces:

```
# taskset -c 0-13 /opt/snort/bin/snort --daq dpdk --daq-var
dpdk_argc="-n4" -i "dpdk0" -z 2
```

Tip: **`taskset`** is used here to pin the cores used by Snort to run only on NUMA node 0. This is done to avoid QPI utilization. Run the cores on the same NUMA node as the NIC.

Related links

Snort site: https://www.snort.org/

Snort source: https://www.snort.org/downloads#snort-downloads

Snort 3 source: https://github.com/snort3/snort3

Snort documentation: https://www.snort.org/documents

4.5 Bro

Installing and configuring Bro for Napatech Link™ Capture Software.

Introduction

Bro is an open source Unix-based network monitoring framework.

Bro can be installed for use with Napatech Link™ Capture Software in two ways:

- Bro can be installed from binary packages and use libpcap interfaces to Napatech Link™ Capture Software.
- To get optimal performance, install a Napatech packet source plugin for Bro.

Package-based installation

Find a suitable precompiled Bro package for your OS distribution. See Installing and Running libpcap Applications on page 42 for general information about how to install and use package-based applications with Napatech libpcap.

See DN-0428 for more information about configuration of Napatech libpcap.

For general information about installation of Bro, see https://www.bro.org/sphinx/install/install.html.

For installation instructions for specific Bro versions, see the `INSTALL` file in the Bro package.

To use the Napatech interfaces, edit the `node.cfg` file according to your specific configuration.

This example sets up a 2-node cluster that listens on the Napatech pcap devices `napa0` and `napa1`. This configuration works with the default `ntpcap.ini` file.

```
# Example BroControl node configuration.
#

# This is a complete standalone configuration.  Most likely you
# only need to change the interface.
# [bro]
# type=standalone
# host=localhost
# interface=napa0

## Below is an example clustered configuration. If you use this,
## remove the [bro] node above.

[manager]
type=manager
host=localhost

[proxy-1]
type=proxy
host=localhost

[worker-1]
type=worker
host=localhost
interface=napa0

[worker-2]
type=worker
host=localhost
interface=napa1
```

Napatech packet source plugin for Bro

Building and running Bro with Napatech support involves the following steps:

- Install Napatech Link™ Capture Software. See Installing Napatech Link™ Capture Software on page 13.
- Install Bro, Bro Package Manager and Bro source.
- Install bro-napatech, using Bro Package Manager.

Related links

Bro site: https://www.bro.org/

Bro source: https://github.com/bro

Bro documentation: https://www.bro.org/documentation/index.html

Bro installation from source: https://www.bro.org/sphinx/install/install.html

Bro binary packages: https://www.bro.org/download/packages.html

Bro Package Manager documentation: http://bro-package-manager.readthedocs.io/en/stable/

Napatech packet source plugin for Bro: https://github.com/hosom/bro-napatech

4.6 TRex

Installing and configuring TRex for use with Napatech Link™ Capture Software.

Introduction

TRex is a fast realistic open source traffic generation tool, running on standard Intel processors. TRex uses DPDK libs and drivers, included in TRex. Napatech Link™ Capture Software provides a kernel driver that DPDK can run on top of, replacing the DPDK kernel driver **`igb_uio`**.

TRex must be built from source.

Steps

To build and run TRex with Napatech support:

1. Download the TRex source: https://github.com/cisco-system-traffic-generator/trex-core, or clone from https://github.com/cisco-system-traffic-generator/trex-core.git.
2. Install Napatech Link™ Capture Software: see Installing Napatech Link™ Capture Software on page 13.
3. Start and stop **`ntservice`** in order to create the driver configuration file `ntservice.ini`:

```
$ sudo /opt/napatech3/bin/ntstart.sh
Loading nt3gd driver                                        [Done]
Creating driver device file                                 [Done]
Loading nt3gd_netdev driver                                 [Done]
Creating driver device file                                 [Done]
Starting NTService (this may take a while)                  [Done]

$ sudo /opt/napatech3/bin/ntstop.sh
Stopping NTService (this may take a while)
NTService stopped                                           [Done]
```

4. Change the number of RX and TX buffers in `ntservice.ini` from 4 to 32:

```
HostBuffersRx = [32,16,-1]                # [x1, x2, x3], ...
HostBuffersTx = [32,16,-1]                # [x1, x2, x3], ...
```

5. Start **`ntservice`**.
6. Build TRex with Napatech support:

```
#  cd linux_dpdk
#./b configure --no-mlx=NO_MLX --with-ntacc
Setting top to                           : /usr/project/trex-core
Setting out to                           :
/usr/project/trex-core/linux_dpdk/build_dpdk
Checking for program 'gcc, cc'           : /usr/bin/gcc
Checking for program 'ar'                : /usr/bin/ar
Checking for program 'g++, c++'          : /usr/bin/g++
Checking for program 'ar'                : /usr/bin/ar
Checking for program 'ldd'               : /usr/bin/ldd
Checking for library z                   : yes
Build sanitized images (GCC >= 4.9.0)    : no
Checking for NTAPI                       : Found needed NTAPI
library
'configure' finished successfully (0.408s)

#./b
 Waf: Entering directory
`/usr/builds/trex-core.github.rsync/linux_dpdk/build_dpdk'
 Info: Using external libverbs.
 update version files
 ...
 [1206/1210] Compiling version.c
 [1207/1210] Linking build_dpdk/linux_dpdk/_t-rex-64-debug-o
 [1208/1210] Linking build_dpdk/linux_dpdk/_t-rex-64-debug
 [1209/1210] Linking build_dpdk/linux_dpdk/_t-rex-64-o
 [1210/1210] Linking build_dpdk/linux_dpdk/_t-rex-64
 Waf: Leaving directory
`/usr/builds/trex-core.github.rsync/linux_dpdk/build_dpdk'
*** generating softlinks ***
'build' finished successfully (15.621s)
```

7. Create a TRex configuration file:

   Use `dpdk_setup_ports.py` to create the config file:

```
# ./dpdk_setup_ports.py -i
By default, IP based configuration file is created. Do you want to use MAC based config?
 (y/N)
+----+------+---------+-------------------+------------------------------------------------+--------+------------+----------+
| ID | NUMA |   PCI    |         MAC         |                         Name
        | Driver |   Linux IF   |   Active   |
+====+======+=========+===================+================================================+========+============+==========+
| 0  | 0     | 05:00.0 |                     | NT20E2 Network Adapter 2x10Gb
       | nt3gd   |             |            |
+----+------+---------+-------------------+------------------------------------------------+--------+------------+----------+
| 1  | 0     | 07:00.0 | ac:22:0b:02:6d:21 | 82574L Gigabit Network Connection
      | e1000e | enp7s0      | *Active* |
+----+------+---------+-------------------+------------------------------------------------+--------+------------+----------+
| 2  | 0     | 08:00.0 | ac:22:0b:02:6d:22 | 82574L Gigabit Network Connection
      | e1000e | enp8s0      |            |
+----+------+---------+-------------------+------------------------------------------------+--------+------------+----------+
| 3  | 1    | 82:00.0 | 00:1b:21:77:c5:54 | 82599ES 10-Gigabit SFI/SFP+ Network Connection
 | ixgbe   | enp130s0f0 |            |
+----+------+---------+-------------------+------------------------------------------------+--------+------------+----------+
| 4  | 1    | 82:00.1 | 00:1b:21:77:c5:55 | 82599ES 10-Gigabit SFI/SFP+ Network Connection
 | ixgbe   | enp130s0f1 |            |
+----+------+---------+-------------------+------------------------------------------------+--------+------------+----------+
| 5  | 1     | 86:00.0 |                     | NT100E3-1-PTP Network Adapter 1x100Gb
      | nt3gd   |             |            |
+----+------+---------+-------------------+------------------------------------------------+--------+------------+----------+
| 6  | 1     | 87:00.0 |                     | NT100E3-1-PTP Network Adapter 1x100Gb
      | nt3gd   |             |            |
+----+------+---------+-------------------+------------------------------------------------+--------+------------+----------+
| 7  | 1     | 8a:00.0 |                     | NT80E3-2-PTP Network Adapter 2x40Gb
      | nt3gd   |             |            |
+----+------+---------+-------------------+------------------------------------------------+--------+------------+----------+
| 8  | 1     | 8b:00.0 |                     | NT80E3-2-PTP Network Adapter 2x40Gb
      | nt3gd   |             |            |
+----+------+---------+-------------------+------------------------------------------------+--------+------------+----------+
Choose even number of interfaces from the list above, either by ID , PCI or Linux IF
Stateful uses order of interfaces: Client1 Server1 Client2 Server2 etc. for flows.
Stateless can be in any order.
For performance, try to choose each pair of interfaces to be on the same NUMA.
Enter list of interfaces separated by space (for example: 1 3) : 7 8

For interface 7, assuming loopback to it's dual interface 8.
Putting IP 1.1.1.1, default gw 2.2.2.2 Change it?(y/N).
For interface 8, assuming loopback to it's dual interface 7.
Putting IP 2.2.2.2, default gw 1.1.1.1 Change it?(y/N).
Print preview of generated config? (Y/n)
### Config file generated by dpdk_setup_ports.py ###

 - port_limit: 2
   version: 2
   interfaces: ['8a:00.0', '8b:00.0']
   port_bandwidth_gb: 40
   port_info:
       - ip: 1.1.1.1
         default_gw: 2.2.2.2
       - ip: 2.2.2.2
         default_gw: 1.1.1.1

   platform:
       master_thread_id: 0
       latency_thread_id: 1
       dual_if:
         - socket: 1
           threads: [8,9,10,11,12,13,14,15,24,25,26,27,28,29,30,31]
```

```
Save the config to file? (Y/n)
Default filename is /etc/trex_cfg.yaml
Press ENTER to confirm or enter new file:
File /etc/trex_cfg.yaml already exist, overwrite? (y/N)Y
Saved to /etc/trex_cfg.yaml.
```

8. Get the SmartNIC bus ID:

   To find the correct Bus ID, run **`adapterinfo`** [Link].

9. Edit the TRex configuration file to include the Bus ID:

   For example: `dpdk_setup_ports.py` generates the following config file:

```
### Config file generated by dpdk_setup_ports.py ###

 - port_limit: 2
   version: 2
   interfaces: ['8a:00.0', '8b:00.0']
   port_bandwidth_gb: 40
   port_info:
       - ip: 1.1.1.1
         default_gw: 2.2.2.2
       - ip: 2.2.2.2
         default_gw: 1.1.1.1

   platform:
       master_thread_id: 0
       latency_thread_id: 1
       dual_if:
         - socket: 1
           threads: [8,9,10,11,12,13,14,15,24,25,26,27,28,29,30,31]
```

   Edit the line `interfaces: ['8a:00.0', '8b:00.0']` to `interfaces: ['8a:00.0/0', '8a:00.0/1']` in order to use the correct bus ID and port 0 and port 1.

```
### Config file generated by dpdk_setup_ports.py ###

 - port_limit: 2
   version: 2
   interfaces: ['8a:00.0', '8b:00.0']
   port_bandwidth_gb: 40
   port_info:
       - ip: 1.1.1.1
         default_gw: 2.2.2.2
       - ip: 2.2.2.2
         default_gw: 1.1.1.1

   platform:
       master_thread_id: 0
       latency_thread_id: 1
       dual_if:
         - socket: 1
           threads: [8,9,10,11,12,13,14,15,24,25,26,27,28,29,30,31]
```

TRex setup and use

See the TRex documentation for how to set up TRex and Linux with `hugepages` and other settings.

Related links

TRex site: https://trex-tgn.cisco.com/.

TRex source: https://github.com/cisco-system-traffic-generator/trex-core.

TRex documentation: http://trex-tgn.cisco.com/trex/doc/.

Installation guide for Napatech support:

- http://trex-tgn.cisco.com/trex/doc/trex_appendix_napatech.html or
- Cisco System Traffic Generator on GitHub

5 Reference Information

This chapter contains the following sections:

5.1 Specifications

Supported operating systems

Napatech Link™ Capture Software with Intel® PAC with Intel® Arria® 10 GX FPGA is validated for and supported on the following platforms:

Linux Distribution	Versions
Ubuntu, 64 bit	16.04
CentOS, 64 bit	7.4 or later
Red Hat (RHEL), 64 bit	7.4 or later

libpcap versions

The Napatech Software Suite includes following versions of libpcap, extended for use with Napatech software:

- libpcap 1.8.1
- libpcap 1.7.3
- libpcap 1.6.2

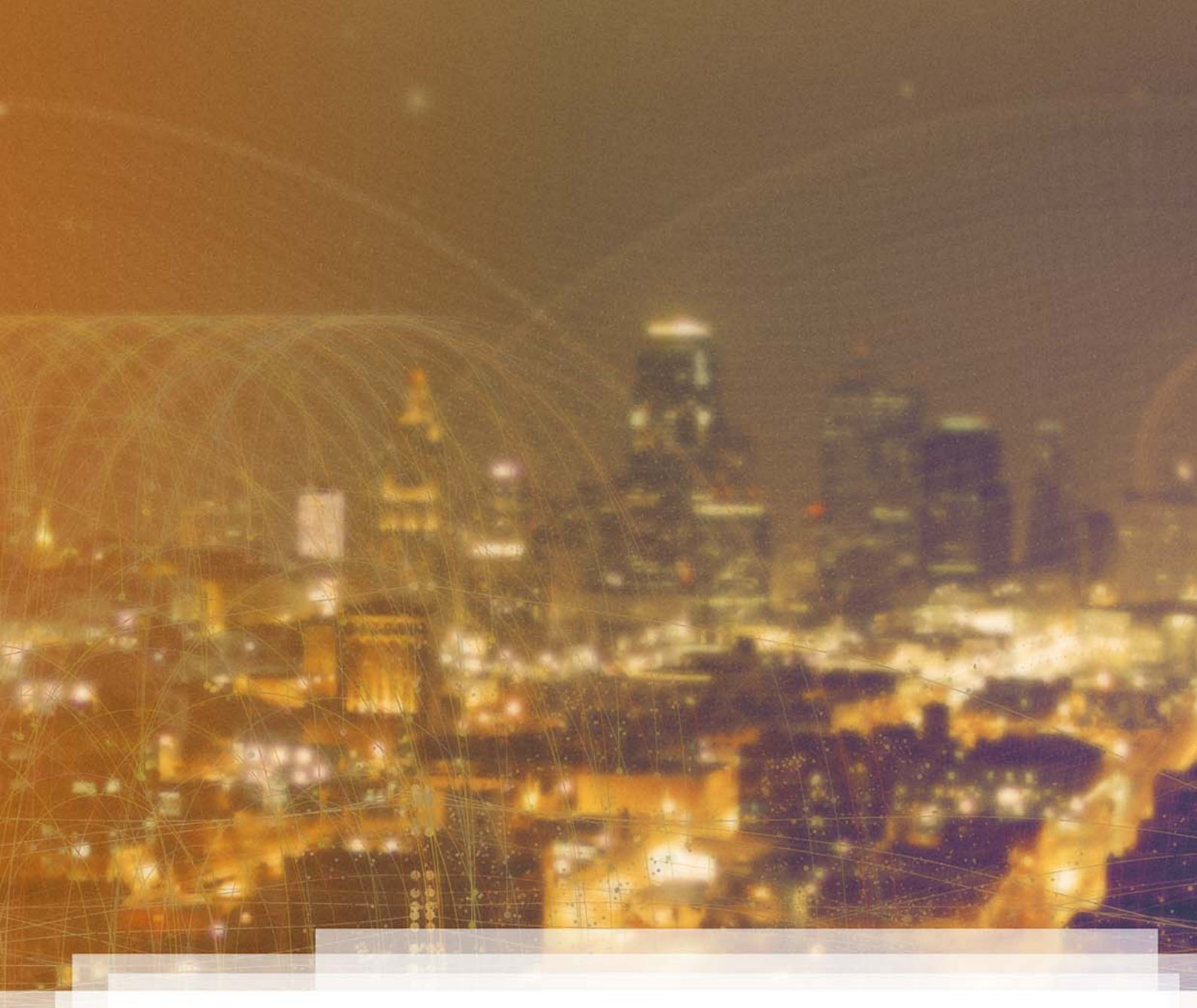
5.2 Default Configuration

The default configurations created by the **`setup_wizard`** tool.

Setup Wizard default configurations

When you press **R** in the **`setup_wizard`** tool to reset the configuration, the following default values are used:

Setting	libpcap		NTAPI
	No ports merged across SmartNICs	Ports merged across SmartNICs	
Interface/stream configuration (port merging and packet distribution)	One libpcap interface per port		One NTAPI stream per port
Tx capable interfaces	Not load distributed: One Tx buffer per libpcap interface Load distributed: No Tx buffers assigned		
Packet distribution to interfaces/streams	Round robin		Round robin
Buffer size for Rx interfaces	16 MB		16 MB
Synchronize SmartNIC time to external reference (OS Time)	Enabled		Enabled
ntservice.ini: NtplFileName	(blank)		/opt/napatech3/config/default.ntpl
ntservice.ini: TimeStampFormat	PCAP_NS	NATIVE_UNIX	NATIVE_UNIX
ntservice.ini: PacketDescriptor	PCAP	NT	NT
ntpcap.ini: NtInterface	Segment	Packet	



Back Matter

The back matter contains these sections:

Style Conventions

Bold typeface is used for names of, for instance, user interface elements and software components.

Italic typeface is used for replaceable text.

`Monospaced typeface` is used for code, commands and file names.

Abbreviations

Abbreviation	Explanation
A10	Arria® 10
ACS	Access Control Services
AFU	Accelerator Functional Unit
BIOS	Basic Input/Output System
CentOS	Community ENTerprise Operating System
CPU	Central Processing Unit
DAQ	Data Acquisition
DN	Document Number
DEB	A .deb file is a Debian-based distribution file.
DPDK	Data Plane Development Kit
FPGA	Field-Programmable Gate Array
GTP	GPRS Tunneling Protocol
HW	HardWare
I	Input
ID	IDentifier
IDS	Intrusion Detection System
IP	Internet Protocol
IPF	IP Fragment handling
IPS	Intrusion Prevention System
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
LAN	Local Area Network
libpcap	Packet CAPture LIBrary
nt, NT	NapaTech
NTAPI	NapaTech Application Programming Interface

Abbreviation	Explanation
ntpl, NTPL	NapaTech Programming Language
NUMA	Non-Uniform Memory Access
NVGRE	Network Virtualization using Generic Routing Encapsulation
O	Output
OISF	Open Information Security Foundation
OPAE	Open Programmable Acceleration Engine
OS	Operating System
PAC	Programmable Acceleration Card
PCIe	Peripheral Component Interconnect Express
PPA	Personal Package Archive
QPI	QuickPath Interconnect
QSFP+	Quad Small Form-factor Pluggable Plus
Rev.	REVision
RHEL	Red Hat Enterprise Linux
RPM	RPM Package Manager
RSS	Receive Side Scaling
Rx	Receive
Tx	Transmission/Transmit
UEFI	Unified Extensible Firmware Interface
VLAN	Virtual Local Area Network
VxLAN	Virtual Extensible LAN

References

Document Number	Explanation
DN-0428	Napatech, SmartNICs with Napatech Software Suite, Napatech libpcap Installation, Software Installation Guide
DN-0449	Napatech, Napatech Software Suite, Reference Documentation
DN-0666	Napatech, Napatech Software Suite, Running Snort, Application Note
DN-1128	Napatech, 4GA SmartNICs with Napatech Software Suite, Feature Set N-ANL11, Feature Description
Intel-UG-20064	Intel Acceleration Stack Quick Start Guide for Intel Programmable Acceleration Card with Intel Arria 10 GX FPGA

Index