napa:tech;
RECONFIGURABLE COMPUTING

Release Summary

Link™ Capture Software 12.7

This document gives an overview of the functionality of the Link™ Capture Software 12.7 release for the Napatech SmartNICs as well as the Intel® Programmable Acceleration Card with Intel® Arria® 10 GX FPGA.

napa:tech;

Modification History

This document has been updated as follows:

Rev.	Date	Comment
1	2021-05-20	Initial version.
2	2021-06-21	Updated according to the latest release.

Contents

1   Summary of the Link™ Capture Software 12.7 Release

In this document

This document gives a brief overview of the Link™ Capture Software 12.7 release for the NT SmartNICs as well as the Intel® Programmable Acceleration Card with Intel® Arria® 10 GX FPGA (Intel® PAC with Intel® Arria® 10 GX FPGA or Intel® PAC A10 GX for short).

Terminology

In this document the term NT80E3-2-PTP does not include NT80E3-2-PTP-8×10/2×40 SmartNICs, except in the title of the Hardware Installation Guide (DN-0980), the term NT80E3-2-PTP-NEBS does not include NT80E3-2-PTP-NEBS-8×10/2×40 SmartNICs, except in the title of the Hardware Installation Guide (DN-0981). However, the term NT100E3-1-PTP includes NT100E3-1-PTP-NEBS SmartNICs, the term NT80E3-2-PTP includes NT80E3-2-PTP-NEBS SmartNICs, the term NT80E3-2-PTP-8×10/2×40 includes NT80E3-2-PTP-NEBS-8×10/2×40 SmartNICs, the term NT40E3-4-PTP includes NT40E3-4-PTP-NEBS SmartNICs, and the term NT20E3-2-PTP includes NT20E3-2-PTP-NEBS SmartNICs, except in contexts where the NEBS SmartNICs are also mentioned specifically.

This chapter contains the following sections:

1.1   Contents of the Link™ Capture Software 12.7 Release

Hardware platforms

The Link™ Capture Software 12.7 release applies to these hardware platforms:

- Intel® Programmable Acceleration Card with Intel® Arria® 10 GX FPGA
- NT200A02 SmartNICs
- NT200A01 SmartNICs
- NT100A01 SmartNICs
- NT100E3-1-PTP SmartNICs
- NT80E3-2-PTP SmartNICs
- NT80E3-2-PTP-8×10/2×40 SmartNICs
- NT50B01 SmartNICs
- NT40A01 SmartNICs
- NT40E3-4-PTP SmartNICs
- NT20E3-2-PTP SmartNICs

Operating systems

The Link™ Capture Software 12.7 release supports:

- Linux kernel 3.x, 4.x and 5.x, 64-bit

  Note: For Intel® PAC with Intel® Arria® 10 GX FPGA only Linux kernel 3.10 through 4.7 are supported.

- Windows Server 2016 and Windows Server 2019, 64-bit

Note: NT50B01 and Intel® PAC with Intel® Arria® 10 GX FPGA are only supported for Linux.

Note: The 2 × 25/10 Gbit/s image for NT200A01 and the 4 × 10/1 SLB image for NT40A01 are only supported for Linux.

Supported server architectures	These server architectures are supported:
	• Intel® x86 64-bit
	• 2nd Gen AMD EPYC™ 64-bit

New features	These features are new in the Link™ Capture Software 12.7 release:

- Support of a configurable parameter, **SDRAMSplit** in the `ntservice.ini` file for memory split between packet buffering and flow matching on the onboard SDRAM of NT200A02 SmartNICs running on a flow management image and NT100A1 SmartNICs running on a flow management image.
- Support for a 2 × 25/10 Gbit/s and a 2 × 10/1 Gbit/s test and measurement image on NT50B01.
- Conditional and generic header stripping of received frames on NT100A01 SmartNICs running on a test and measurement image.
- Conditional packet masking of received frames based on dynamic and fixed offsets on NT100A01 SmartNICs running on a test and measurement image.
- VLAN tagging of frames to be locally retransmitted to enable load balancing on NT100A01 SmartNICs running on a test and measurement image.

Napatech Link™ Capture Software features	The general features of Napatech Link™ Capture Software are described in Napatech Link™ Capture Software Features on page 8.

Product packages	The Link™ Capture Software 12.7 release consists of these product packages:

- **ntanl_package_3gd-12.7.x-linux**
- **ntanl_package_3gd-12.7.x-windows**

The **ntanl_package_3gd-12.7.x-linux** product package includes FPGA images for the NT SmartNICs (with **imgctrl** tool for upgrading), two Accelerator Functional Units (AFUs) for the Intel® Programmable Acceleration Card with Intel® Arria® 10 GX FPGA for running at 1 × 40 Gbit/s and at 4 × 10 Gbit/s, respectively, driver software, tools packages, libpcap and documentation. The software components are described in Software on page 16.

The **ntanl_package_3gd-12.7.x-windows** product package includes FPGA images for the NT SmartNICs (with **imgctrl** tool for upgrading), driver software, tools packages, WinPcap and documentation. The software components are described in Software on page 16.

Included documents

The Link™ Capture Software 12.7 release includes these documents:

Note: All of these documents are not necessarily included in each product package.

Note: The documents can also be found on docs.napatech.com (the Napatech Documentation Portal).

- Napatech, NT SmartNICs with Napatech Link™ Capture Software, Getting Started, Overview Document, DN-0931 (**UPDATED**)
- Napatech, Napatech Link™ Capture Software for Intel® PAC with Intel® Arria® 10 GX FPGA, Installation and Use, Quick Guide, DN-1113
- Napatech, Using the Napatech Support Portal, User Guide, DN-1118
- Napatech, Link™ Capture Software 12.7, Release Summary, DN-1299 (**NEW**)
- Napatech, From Link™ Capture Software 12.6 to Link™ Capture Software 12.7, Migration Document, DN-1300 (**NEW**)
- Napatech, From Green Bay 2 to Link™ Capture Software 12.7, Migration Document, DN-1301 (**NEW**)
- Napatech, NT200A02-SCC, Installation and Specifications, Hardware Installation Guide, DN-1169
- Napatech, NT200A02-NEBS, Installation and Specifications, Hardware Installation Guide, DN-1200
- Napatech, NT200A01-SCC, Installation and Specifications, Hardware Installation Guide, DN-1016
- Napatech, NT200A01-NEBS, Installation and Specifications, Hardware Installation Guide, DN-1017
- Napatech, NT100A01-SCC, Installation and Specifications, Hardware Installation Guide, DN-1261
- Napatech, NT100A01-NEBS, Installation and Specifications, Hardware Installation Guide, DN-1262
- Napatech, NT100E3-1-PTP, Installation and Specifications, Hardware Installation Guide, DN-0848
- Napatech, NT100E3-1-PTP-NEBS, Installation and Specifications, Hardware Installation Guide, DN-0896
- Napatech, NT80E3-2-PTP, Installation and Specifications, Hardware Installation Guide, DN-0980
- Napatech, NT80E3-2-PTP-NEBS, Installation and Specifications, Hardware Installation Guide, DN-0981
- Napatech, NT50B01, Installation and Specifications, Hardware Installation Guide, DN-1251
- Napatech, NT40A01-SCC, Installation and Specifications, Hardware Installation Guide, DN-0954
- Napatech, NT40A01-NEBS, Installation and Specifications, Hardware Installation Guide, DN-0955
- Napatech, NT40E3-4-PTP, Installation and Specifications, Hardware Installation Guide, DN-0797
- Napatech, NT40E3-4-PTP-NEBS, Installation and Specifications, Hardware Installation Guide, DN-0897
- Napatech, NT20E3-2-PTP, Installation and Specifications, Hardware Installation Guide, DN-0904
- Napatech, NT20E3-2-PTP-NEBS, Installation and Specifications, Hardware Installation Guide, DN-0905
- Napatech, Napatech Link™ Software, Software Installation for Linux, Software Installation Guide, DN-0379
- Napatech, Napatech Link™ Capture Software, Software Installation for Windows and WinPcap-NT Installation, Software Installation Guide, DN-1067 (**UPDATED**)

- Napatech, Napatech Link™ Capture Software, Napatech libpcap Installation, Software Installation Guide, DN-0428 (**UPDATED**)
- Napatech, Napatech Link™ Capture Software, Software Architecture, Overview Document, DN-0810 (**UPDATED**)
- Napatech, Napatech Software Suite, Reference Documentation, DN-0449 (**UPDATED**)
- Napatech, Napatech Link™ Software, Handling FPGA Images, User Guide, DN-0487
- Napatech, SmartNICs with Napatech Link™ Capture Software, Basic Troubleshooting, User Guide, DN-0737
- Napatech, Napatech Link™ Capture Software, Time-Stamping and Time Synchronization, User Guide, DN-0985 (**UPDATED**)
- Napatech, Napatech Link™ Capture Software, Statistics, User Guide, DN-0986
- Napatech, Napatech Link™ Capture Software, Multi-CPU Distribution, User Guide, DN-0988
- Napatech, Napatech Link™ Capture Software, Stateful Flow Management, User Guide, DN-1227 (**UPDATED**)
- Napatech, Napatech Link™ Capture Software, Running Snort on Linux, Application Note, DN-0666
- Napatech, Napatech Link™ Capture Software, Opening Multiple Network Streams as One, Application Note, DN-0740
- Napatech, NT SmartNICs with Napatech Link™ Capture Software, Qualifying, Fine-Tuning and Troubleshooting Time Synchronization, Application Note, DN-0793
- Napatech, NT80E3-2-PTP, Running at 8 × 10 Gbit/s, Application Note, DN-1133
- Napatech, Napatech Link™ Capture Software Features, Feature Description, DN-1128 (**UPDATED**)

1.2   Napatech Link™ Capture Software Features

Data sharing

Multiple streams can get access to the same host buffer.

Packet merging

Packets from different host buffers can be merged into one stream based on time stamps. In this way traffic from different SmartNICs can be merged.

Port virtualization

All ports on all SmartNICs are numbered sequentially.

Hardware abstraction

HAL (hardware abstraction layer) macros are provided for all hardware-generated information in the standard packet descriptor and in extended descriptor 9.

1.3   4GA Feature Overview

In this section

This section gives an overview of the 4GA features in the Link™ Capture Software 12.7 release.

Packet descriptors

These packet descriptors are available:

- Standard packet descriptor
- Dynamic packet descriptor 1
- Dynamic packet descriptor 2
- Dynamic packet descriptor 3
- Dynamic packet descriptor 4
- PCAP packet descriptor

Note: Dynamic packet descriptor 4 does not apply to NT40A01 SmartNICs running on the 4 × 10/1 Gbit/s SLB image.

Packet descriptor extension	For frames with a standard packet descriptor, extended packet descriptor 9 is available.
Statistics counters per port	RMON1 and extended RMON1 statistics counters are available for each port.

Statistics counters per color These statistics counters are available for each color:

- Frame counters
- Byte counters

Statistics counters per stream ID These statistics counters are available for each stream ID:

- Frame counters for forwarded frames
- Frame counters for flushed frames
- Frame counters for dropped frames
- Byte counters for forwarded frames
- Byte counters for flushed frames
- Byte counters for dropped frames

Port merging Frames from multiple ports on the same SmartNIC can be merged into one stream.

QPI bypass When two NT100E3-1-PTP SmartNICs are connected and working as a pair, it is possible to transfer data received on one NT100E3-1-PTP SmartNIC via the interconnect cable to the other NT100E3-1-PTP SmartNIC and vice versa. In this way data destined for a specific NUMA node can be delivered over the PCIe bus of the SmartNIC that is local to this NUMA node and thereby the QPI in the host can be bypassed to avoid introducing additional latency.

CPU socket load-balancing Two NT40A01 SmartNICs running on the 4 × 10/1 Gbit/s SLB image, or two NT200A02 SmartNICs running on the 2 × 40 Gbit/s SLB image or on the 8 × 10 Gbit/s SLB image can be bonded in a master/slave configuration so that all traffic received on the master SmartNIC is replicated to the slave SmartNIC, ensuring local cache access to two NUMA nodes in a dual-CPU-socket server to optimize performance.

Time stamp formats Received frames are time-stamped according to the internal SmartNIC time either when the first byte is received or when the last byte is received.

Note: Start-of-frame time-stamping does not apply to NT40E3-4-PTP SmartNICs running on the capture/replay image, to NT40A01 SmartNICs running on the 4 × 10/1 Gbit/s SLB image nor to the Intel® Programmable Acceleration Card with Intel® Arria® 10 GX FPGA.

These time stamp formats are available:

- PCAP-ns format, 1 ns (only if PCAP packet descriptor is selected)
- Native UNIX format, 1 ns (not for NT40A01 running on the 4 × 10/1 Gbit/s SLB image)
- Native UNIX format, 10 ns
- PCAP-µs format, 1000 ns (only if PCAP packet descriptor is selected)

All received frames use the same time stamp format. Offset compensation is available.

RX data path delay values are available in the API to allow applications to compensate for these delays.

Time stamp synchronization An NT50B01 SmartNIC and an Intel® Programmable Acceleration Card with Intel® Arria® 10 GX FPGA can be synchronized to OS time with dynamic drift adjustment.

Note: The rest of the time stamp synchronization information does not apply to NT50B01 SmartNICs nor to the Intel® Programmable Acceleration Card with Intel® Arria® 10 GX FPGA

Two or more NT SmartNICs can be synchronized, one being the master and the other(s) being slaves connected using daisy chain.

When configured as master and connected to another NT SmartNIC, an NT SmartNIC can provide a synchronized set of external time and SmartNIC time every second. The external time is the time of the other SmartNIC. Synchronized sets of external time and SmartNIC time can also be obtained every 20 µs.

When an NT SmartNIC is configured as master and connected to a time device, for instance a GPS, the PPS signal from the time device can trigger the sampling of the SmartNIC time.

The NT SmartNIC time stamp clock rate can be synchronized relative to any external time source with a PPS output at TTL levels. If absolute time information is available, the SmartNIC time clock can be synchronized to the absolute UTC time.

Minor adjustments to the internal NT SmartNIC time can be done using a sliding adjust; and cable delays can be compensated for.

When two or more NT SmartNICs are synchronized with respect to absolute time, block statistics are transferred synchronously from the SmartNICs to the host.

For all time reference sources (with OS, NT-TS, PPS or PTP time synchronization) the same time synchronization statistics are supported for NT SmartNICs:

- Current offset to master
- Mean value
- Minimum and maximum offset to master
- Peak-to-peak jitter
- Calculated mean offset to master
- Calculated standard deviation
- Time since last reset of statistics calculation

IEEE 1588-2008 PTP v2 clock synchronization The IEEE 1588-2008 PTP v2 clock synchronization functionality allows the SmartNICs to be synchronized against a PTP grandmaster, using the PTP Ethernet port on the SmartNIC.

Note: IEEE 1588-2008 PTP v2 clock synchronization does not apply to NT50B01 SmartNICs nor to the Intel® Programmable Acceleration Card with Intel® Arria® 10 GX FPGA.

These PTP Ethernet port configurations are supported: Static IP, DHCP and VLAN.

These communication protocols are supported: IPv4/UDP and IEEE802.3 (layer2).

These PTP profiles are supported: PTP Default, Telecom, Power, Enterprise and G.8275.1.

The SmartNICs support both end-to-end (delay request-response mechanism) and peer-to-peer (peer delay mechanism) for propagation delay measurements.

SyncE support NT SmartNICs can utilize SyncE-enabled networks to provide highly stable frequency synchronization.

Note: SyncE does not apply to NT200A02, NT200A01, NT100A01 and NT50B01 SmartNICs nor to the Intel® Programmable Acceleration Card with Intel® Arria® 10 GX FPGA.

Frame classification	Frames are inspected and classified by the frame decoder. These protocols are identified for the different layers:

- Layer 2: ISL encapsulation, VLAN tags, MPLS encapsulation, Cisco FabricPath (CFP) encapsulation (not for NT40E3-4-PTP running on the capture/replay image nor NT40A01 running on the 4 × 10/1 Gbit/s SLB image), VN-Tag classification (not for NT40E3-4-PTP running on the capture/replay image nor NT40A01 running on the 4 × 10/1 Gbit/s SLB image)
- Layer 2: EtherII, Novell_RAW, SNAP, LLC, others
- Layer 3: IPv4, IPv6, others
- Layer 4: TCP, UDP, ICMP, SCTP, others
- Tunnel type: GTPv0-U, GTPv1-C, GTPv2-C, GTPv1-U, GRE_v0 (including NVGRE), GRE_v1, IPinIP, EtherIP, VXLAN, others
- Inner layer 2: VLAN tags, MPLS encapsulation
- Inner layer 3: IPv4, IPv6, others
- Inner layer 4: TCP, UDP, ICMP, GRE_v0, SCTP, others

Frames can also be classified as small, large or jumbo frames.

Note: IPv6 is generally supported to the same level as IPv4. In addition, the frame decoder supports a broad range of IPv6 extension headers, as well as filtering based on IPv6 addresses.

Layer 3 and layer 4 checksums can be verified on NT200A02, NT100A01 and NT50B01 (**NEW**) SmartNICs running on a test and measurement image.

Filtering	Filtering can be based on:

- Port numbers
- Pattern compares
- Protocol information
- Frame size tests
- Frame error tests
- Key matching (not for NT200A02 running on the 2 × 100 Gbit/s FM1 image, on the 2 × 40 Gbit/s FM image nor on the 8 × 10 Gbit/s FM image, for NT100A01 running on the 4 × 25/10 Gbit/s FM1 image or on the 4 × 10/1 Gbit/s FM1 image, nor for NT40E3-4-PTP SmartNICs running on the capture/replay 2 image)
- IP address matching including address groups and wildcard matches (not for NT200A02 running on the 2 × 100 Gbit/s FM1 image, on the 2 × 40 Gbit/s FM image nor on the 8 × 10 Gbit/s FM image, nor for NT100A01 running on the 4 × 25/10 Gbit/s FM1 image nor on the 4 × 10/1 Gbit/s FM1 image)
- User-defined key tests that include arbitrary fields

Overlapping filters can be prioritized.

Filtering on tunneled IP traffic is available for GTPv0-U, GRE_v0, IPinIP and EtherIP tunnels.

Packet Coloring	The packet coloring functionality enables tagging of captured frames with a color ID based on the filter logic. The color ID can contain contributions from one filter with the highest priority (color) and/or contributions from a number of filters that the frame matches (color mask). Packet coloring can be used, for instance, in connection with multi-CPU distribution (see Multi-CPU distribution on page 13).

Correlation of Packets	Correlation keys can be generated to enable quality of service measurements between multiple points in a network and to accelerate packet deduplication in application software.

Note: Correlation of packets does not apply to NT200A02 running on the 2 × 100 Gbit/s FM1 image, on the 2 × 40 Gbit/s FM image nor on the 8 × 10 Gbit/s FM image, to NT200A01 running at 2 × 100 Gbit/s, to NT100A01 running on the 4 × 25/10 Gbit/s FM1 image nor on the 4 × 10/1 Gbit/s FM1 image, nor to NT40A01 running on the 4 × 10/1 Gbit/s SLB image.

Deduplication	The deduplication functionality enables discarding or retransmission of duplicate or nonduplicate frames. Frames are considered to be duplicate frames if they (or a part of them) have the same correlation key, they are not separated by more than a specified time, and if they belong to the same specified part of the traffic. The deduplication functionality generates per-port statistics over the number of frames being discarded, retransmitted and detected as duplicates.

Protocol offsets and masking settings can be used to determine which parts of the frames are compared.

Note: Deduplication does not apply to NT200A02 running on the 2 × 100 Gbit/s FM1 image, on the 2 × 40 Gbit/s FM image nor on the 8 × 10 Gbit/s FM image, to NT200A01 running at 2 × 100 Gbit/s nor 2 × 25 Gbit/s, to NT100A01 running on the 4 × 25/10 Gbit/s FM1 image nor on the 4 × 10/1 Gbit/s FM1 image, to NT100E3-1-PTP, to NT40E3-4-PTP running on the capture/replay image, nor to NT40A01 running on the 4 × 10/1 Gbit/s SLB image.

Slicing	Slicing can be both dynamic and fixed, or disabled. These slicing modes are available:

- Fixed length
- Fixed length + ISL
- Fixed length + ISL + ETH + VLAN
- Fixed length + ISL + ETH + VLAN + MPLS
- Fixed length + ISL + ETH + VLAN + MPLS + L3
- Fixed length + ISL + ETH + VLAN + MPLS + L3 + L4
- Fixed length + ISL + ETH + VLAN + MPLS + L3 + L4 + outer data type
- Fixed length + ISL + ETH + VLAN + MPLS + L3 + L4 + outer data type + inner L3
- Fixed length + ISL + ETH + VLAN + MPLS + L3 + L4 + outer data type + inner L3 + inner L4
- End of frame

The end-of-frame dynamic offset enables bytes to be sliced off from the end of the frame by applying a negative offset. This can be used, for instance, for frame checksum removal.

Header stripping	Header stripping of received frames can be both conditional and generic. Received frames can be stripped of the tunneling protocols that can be recognized by the frame decoder, such as GRE, GTP, NVGRE, VXLAN and Cisco FabricPath as well as of the tags that can be recognized by the frame decoder, such as VLAN, MPLS and VN-Tag. In addition a portion of a received frame can be stripped based on an offset and a length.

Note: Header stripping only applies to NT200A02 running on a capture/replay image and NT100A01 running on a test and measurement image (**NEW**).

Packet masking	A specific part of a packet payload based on a dynamic plus a static offset and a length can be masked out by zeroing it.

Note: Packet masking only applies to NT200A02 running on a capture/replay image and NT100A01 running on a test and measurement image (**NEW**).

Hash value generation

Hash values can be generated from many types of hash keys based on packet header information:

- Last MPLS label
- All MPLS labels
- 2-tuple
- 2-tuple, sorted
- Last VLAN ID
- All VLAN IDs
- 5-tuple
- 5-tuple, sorted
- 3-tuple GREv0
- 3-tuple GREv0, sorted
- 5-tuple SCTP
- 5-tuple SCTP, sorted
- 3-tuple GTPv0
- 3-tuple GTPv0, sorted
- 3-tuple GTPv1 or GTPv2
- 3-tuple GTPv1 or GTPv2, sorted
- Inner 2-tuple
- Inner 2-tuple, sorted
- Inner 5-tuple
- Inner 5-tuple, sorted
- IP fragment tuple
- Round-robin

Hash keys can be selected dynamically for different types of frames.

Source and destination addresses and ports can be swapped in hash calculations. Hash swapping can be based on inner and/or outer IP match lists specifying certain IP addresses.

Multi-CPU distribution (see Multi-CPU distribution on page 13) can be controlled using hash key masks. Hash word bits that are masked out are set to 0. In this way certain parts of the input data can be disregarded from the hash calculation, so that frames with hash values that only differ due to, for instance, port numbers can be configured to end up in the same host buffer.

Multi-CPU distribution

Multi-CPU distribution enables the SmartNIC to off-load the CPU load-balancing by distributing the processing of captured frames in the host CPU. Data can be placed in separate buffers based on port numbers, hash values and filtering.

IP fragment handling

The IP fragment handling functionality accelerates the processing of fragmented IP traffic and enables the use of 5-tuple (or other non 2-tuple) hash keys on fragments for better CPU load distribution.

Note: IP fragment handling does not apply to NT200A01 running at 2 × 100 Gbit/s nor to NT40E3-4-PTP SmartNICs running on the capture/replay 2 image.

Stateful flow management

The stateful flow management functionality can recognize received frames belonging to specific flows, and apply the same action to these frames, while statistics about the flow are being updated in a flow record. The frames can, for instance, be forwarded to a specific host Rx queue, dropped or fast forwarded to a network port.

The frames are decoded and lookups are made in a flow table to recognize the frames. The flow table can be updated according to learned flows. Flows can be based on several individual fields from the packets, for instance as 5-tuples, 4-tuples, 3-tuples, 2-tuples or combinations.

Flows can be terminated based on TCP flow terminations, timeouts or application requests.

Stateful flow management can, for instance, be used to offload security applications by processing frames from known flows entirely on the SmartNIC and only forwarding frames that are not recognized to the application.

Note: Stateful flow management only applies to NT200A02 running on the 2 × 100 Gbit/s FM1 image, on the 2 × 40 Gbit/s FM image or on the 8 × 10 Gbit/s FM image, and to NT100A01 running on the 4 × 25/10 Gbit/s FM1 image or on the 4 × 10/1 Gbit/s FM1 image.

Local retransmission

The local retransmission functionality enables frames received on one network port to be retransmitted to the same port or to another network port on the same SmartNIC without involving the host CPU. The retransmitted frames can be expanded to include a trailer containing a 64-bit RX time stamp with a resolution of 1 ns.

In NT200A02 SmartNICs running on a capture/replay image and NT100A01 SmartNICs running on a test and measurement image (**NEW**), VLAN tags can be inserted into frames to be locally retransmitted to enable load balancing, and frames can be sliced before being locally retransmitted.

Note: Local retransmission does not apply to NT200A01 running at 2 × 100 Gbit/s.

Line loopback

Frames received on one network port can be retransmitted to the same network port without involving the host CPU. The line loopback functionality and the filtering/capturing functionality can be used independently of each other.

Host-based transmission	The full host-based transmission functionality enables high-speed transmission with low CPU load of frames located in a host buffer in the server application memory. Frames that have been received by a SmartNIC can be retransmitted by the same or a different SmartNIC without modification.

Note: The full host-based transmission functionality does not apply to NT200A01 running on the 2 × 100 Gbit/s capture image nor to NT40E3-4-PTP running on the capture image.

Transmission can be both static to a single port and dynamic, where the application can assign different TX ports for different frames.

On NT200A02, NT100A01 and NT50B01 (**NEW**) SmartNICs running on a test and measurement image it can be specified per packet if the frames are to be transmitted with a new correct CRC, a new incorrect CRC or the stored CRC left untouched. Other SmartNICs always generate a new Ethernet CRC for frames to be transmitted.

Layer 3 and layer 4 checksums can be generated on NT200A02, NT100A01 and NT50B01 (**NEW**) SmartNICs running on a test and measurement image.

A time stamp can be injected into a frame to be transmitted at a specified offset on NT200A02, NT100A01 and NT50B01 (**NEW**) SmartNICs running on a test and measurement image. The offset can be calculated from the start of the frame, from the end of the frame, from the start of the layer 3 header or from the start of the layer 4 header. The value of the time stamp can be the time when the first byte of the frame is put on the wire, or the time when the last byte is put on the wire.

User data, such as VLAN tags can be inserted into the transmitted frames using dynamic descriptor 3.

Transmission can be timed so that frames are transmitted at specific points in time. In this way frames can be transmitted, for instance, according to their RX time stamps, so that they are replayed as captured. Timed transmission also allows synchronized replay of traffic from a number of different SmartNICs when their time stamp clocks are synchronized.

Limited host-based transmission	Limited host-based transmission only applies to NT200A01 running on the 2 × 100 Gbit/s capture image and to NT40E3-4-PTP running on the capture image. This host-based transmission is very CPU-intensive and has a very limited TX rate.
Buffer system	The buffer system supports the use of up to 128 RX buffers per SmartNIC in host memory with dynamic host buffer segment size and up to 64 RX buffers with static host buffer segment size.

Note: Dynamic segment size does not apply to the Intel® Programmable Acceleration Card with Intel® Arria® 10 GX FPGA.

1.4 Software

Software APIs

These software APIs are included:

- NTAPI (Napatech application programming interface)

  Napatech application programming interface is a stream-based, packet- or segment-based zero-copy API. User space libraries are included in binary, and the kernel space driver is included as open source (GPL-compliant).

- NTPL (Napatech programming language)

  Napatech programming language is a highly efficient interface, which optimizes the use of the FPGA resources. In the Link™ Capture Software 12.7 release these functions are supported:

  - Dynamic packet descriptors
  - Filtering
  - Stateful flow management (only applies to NT200A02 running on the 2 × 100 Gbit/s FM1 image, on the 2 × 40 Gbit/s FM image or on the 8 × 10 Gbit/s FM image, and to NT100A01 running on the 4 × 25/10 Gbit/s FM1 image or on the 4 × 10/1 Gbit/s FM1 image)
  - Slicing
  - Hash key definition
  - Multi-CPU distribution
  - Correlation key (does not apply to NT200A01 running at 2 × 100 Gbit/s)
  - Deduplication (does not apply to NT200A01 running at 2 × 100 Gbit/s nor 2 × 25 Gbit/s, to NT100E3-1-PTP, to NT40E3-4-PTP running on the capture/replay image nor to NT40A01 running on the 4 × 10/1 Gbit/s SLB image)
  - Local retransmission (does not apply to NT200A01 running at 2 × 100 Gbit/s)
  - IP fragment handling (does not apply to NT200A01 running at 2 × 100 Gbit/s)

- libpcap

  A libpcap source library is available for integration of libpcap applications.

- WinPcap

  A WinPcap source library is available for integration of WinPcap applications.

Tools

The Link™ Capture Software 12.7 release includes a tools package with a number of binary tools that demonstrate the SmartNIC functionality. These tools are included:

- `adapterinfo` (displays information about the SmartNIC)
- `capfileconvert` (converts captured files to or from Napatech Link™ Capture Software format)
- `capfiledump` (dumps a Napatech Link™ Capture Software capture file in humanly readable format)
- `capture` (captures data to disk)
- `config` (configures links, time stamp values and sensor alarm levels)
- `diagnostics` (perform diagnostics on a SmartNIC)
- `latency` (not used)
- `monitoring` (queries the system port statistics)
- `nimconfig` (reads/writes values from/to the memory of network interface modules)
- `ntinfo` (collects information about the host buffer utilization once per second, and creates a host buffer utilization report)
- `ntlog` (reads out the driver log)
- `ntpcap_capture` (captures data received from a PCAP interface and stores it to disc in true PCAP file format.)
- `ntpcap_replay` (replays any captured PCAP or PCAPNG file)
- `ntpl` (sends NTPL code to a certain network stream)
- `osmode` (creates a virtual network device on top of Napatech Link™ Capture Software)
- `pktgen` (performs traffic generation – does not apply to NT200A01 running on the 2 × 100 Gbit/s capture image nor to NT40E3-4-PTP running on the capture image)
- `pps_basic` (synchronizes the SmartNIC against a PPS signal – does not apply to NT50B01 nor to Intel® PAC with Intel® Arria® 10 GX FPGA)
- `pps_endrun` (synchronizes the SmartNIC against a PPS signal from an EndRun Præcis Cf CDMA receiver or an EndRun Præcis II CDMA receiver – does not apply to NT50B01 nor to Intel® PAC with Intel® Arria® 10 GX FPGA)
- `pps_oregano` (synchronizes the SmartNIC against a PPS signal from an Oregano SYN1588 PCIe NIC – does not apply to NT50B01 nor to Intel® PAC with Intel® Arria® 10 GX FPGA)
- `pps_os` (synchronizes the SmartNIC against a PPS signal using OS time as input – does not apply to NT50B01 nor to Intel® PAC with Intel® Arria® 10 GX FPGA)
- `pps_ptp` (enables PPS on all SmartNICs configured to run PPS – does not apply to NT50B01 nor to Intel® PAC with Intel® Arria® 10 GX FPGA)
- `pps_symmetricom` (synchronizes the SmartNIC against a PPS signal from a Microsemi SyncServer® – does not apply to NT50B01 nor to Intel® PAC with Intel® Arria® 10 GX FPGA)
- `productinfo` (displays product information about a SmartNIC)
- `profiling` (queries the system for profiling data)
- `ptp_port_config` (reads/writes values of selected PTP port parameters – does not apply to NT50B01 nor to Intel® PAC with Intel® Arria® 10 GX FPGA)
- `readproperty` (queries the properties of Napatech Link™ Capture Software)
- `replay` (transmits a file with captured traffic – does not apply to NT200A01 running on the 2 × 100 Gbit/s capture image nor to NT40E3-4-PTP running on the capture image)
- `setup_wizard` (configures the Napatech Link™ Capture Software)
- `supportinfo` (dumps system information for support purposes)
- `throughput` (captures to server memory and shows the throughput)
- `vpd` (reads/writes VPD (vital product data) user data)

Examples

The Link™ Capture Software 12.7 release includes an examples package for the SmartNICs with a number of code examples that use the SmartNIC functionality to illustrate how user applications can be made. These examples are included:

- `bypass/config` (not used)
- `bypass/info` (not used)
- `bypass/watchdog` (not used)
- `config` (is an example of how to use the configuration stream interface in NTAPI)
- `event` (is an example of how to use the event stream interface in NTAPI)
- `eventMonitor` (is an example of how to monitor events and dump information about them to the screen using the event stream interface in NTAPI)
- `flowmatch/flowmatch_example_main` (is an example that covers a number flow matching functionalities working together – only applies to NT200A02 running on the 2 × 100 Gbit/s FM1 image, on the 2 × 40 Gbit/s FM image or on the 8 × 10 Gbit/s FM image, and to NT100A01 running on the 4 × 25/10 Gbit/s FM1 image or on the 4 × 10/1 Gbit/s FM1 image)
- `hashref/calc_single_hash` (is an example of how to use the hash reference library to calculate hash values for some variants of key data)
- `info` (is an example of how to use the information stream interface in NTAPI)
- `integration/systemd` (contains a unit configuration for systemd)
- `integration/sysv` (contains a System V **init** script enabling **ntservice** to be started and stopped by a System V compatible **init** process)
- `integration/upstart` (contains a job file for **Upstart**, enabling **ntservice** to be controlled by **Upstart**)
- `net/addPacket` (is a legacy example of how to implement a simple VLAN tagging, using limited host-based transmission - only applies to NT100E3-1-PTP and NT40E3-4-PTP)
- `net/analysis` (is an example of how to perform real-time analysis of packets using the file network stream interface in NTAPI)
- `net/capfileconvert` (is an example of how to convert captured files)
- `net/capture` (is an example of how to capture to disk using the RX network stream interface in NTAPI)
- `net/checksum` (is an example of how to transmit packets where layer 3 and layer 4 checksums are recalculated based on control bits in the packet descriptor – only applies to NT200A02, NT100A01 and NT50B01 (**NEW**) SmartNICs running on a test and measurement image)
- `net/hostbuffer_poll` (is an example of how to poll for host buffers attached)
- `net/inline` (not used)
- `net/ipfdemo` (is an example of how to use IP fragment handling – does not apply to NT200A01 running at 2 × 100 Gbit/s)
- `net/netflow` (is an example of how to use the dynamic descriptor 1 to extract net flow information from a packet)
- `net/numa` (is an example of how to assign and use a host buffer on a NUMA mode)
- `net/readcapfile` (is an example of how to read a capture file using NTAPI)
- `net/replay` (not used)
- `net/replay4GA` (is an example of how to replay captured data onto a port using the TX network stream interface in NTAPI – does not apply to NT200A01 running on the 2 × 100 Gbit/s capture image nor to NT40E3-4-PTP running on the capture image)
- `net/replayGS` (not used)
- `net/streamidstatistics` (is an example of how to get stream ID statistics that can be correlated with what the application receives)
- `net/timestamp_inject` (is an example of how to control injection of time stamps and FCS generation per packet using dynamic descriptor 3 – only applies to NT200A02, NT100A01 and NT50B01 (**NEW**) SmartNICs running on a test and measurement image)

- `net/transmit_multifunction` (is an example of how to transmit different packet sizes at different rates on different ports by transmitting data in segments using NTAPI)
- `net/transmit_on_timestamp` (is an example of how to transmit packets on time stamp using the packet interface in NTAPI)
- `net/transmit_on_timestamp_setclock` (is an example of how to transmit a sequence of packets on time stamp multiple times using **TXSETCLOCK** to advance the delta with the duration of a single replay iteration)
- `net/transmit_packet` (is an example of how to transmit packets using the packet interface in NTAPI)
- `net/transmit_pcap` (is an example of how to transmit PCAP packets using the packet interface in NTAPI)
- `net/transmit_segment` (is an example of how to transmit segments using the standard packet descriptor and the TX network stream interface in NTAPI)
- `net/transmit_segment_dyn_descr` (is an example of how to transmit segments using a dynamic packet descriptor and the TX network stream interface in NTAPI)
- `net/vlandemo` (is an example of how to implement a simple VLAN tagging, using dynamic descriptor 3 and the inline transmission capabilities – does not apply to Intel® PAC with Intel® Arria® 10 GX FPGA)
- `pps` (is an example of how to use PPS time synchronization – does not apply to NT50B01 nor to Intel® PAC with Intel® Arria® 10 GX FPGA)
- `sensor` (is an example of how to read sensors)
- `stat` (is an example of how to use the statistics stream interface in NTAPI)
- `statUsage` (is an example of how to use the statistics stream interface in NTAPI to read host buffer usage statistics)