Using Specialized Network Adapters to Improve the Accuracy of Network Analysis in Highly-Utilized Networks

June 17th 2010

Pete Sanders

Application Engineering Director | Napatech Inc.

SHARKFEST **‘10**

Stanford University

June 14-17, 2010

Agenda

Network analysis minimum requirements

Packet time stamping and precision

External time synchronization

Eliminating packet loss

Advanced features of Napatech network adapters

Napatech libpcap support

Demonstration

Network Analysis Minimum Requirements

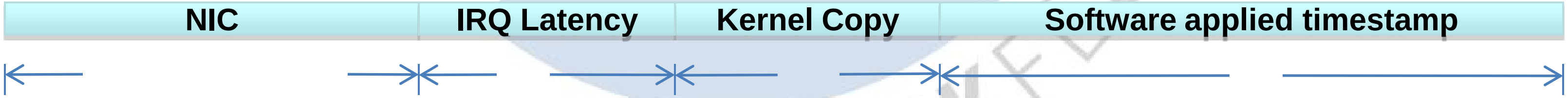
Accurate network analysis requires:

- Accurate time stamping
  - Consistent packet to packet timestamp generation
  - Precision better than minimum frame time on network being analyzed
  - Highly utilizes networks required time stamp precision approaching theoretical minimum frame time: 1Gbps = 700ns, 10Gbps = 70ns
  - Packet fragments: precision requirement may approach minimum Ethernet IFG: 1Gbps = 96ns, 10Gbps = 9.6ns
- Zero packet loss
  - Zero frames dropped by network interface
  - Zero frames dropped by kernel
- Capture all frames including:
  - Frames failing CRC and checksums
  - Packet fragments

Packet time stamping and precision

Standard network interfaces provide non-deterministic time stamping behavior

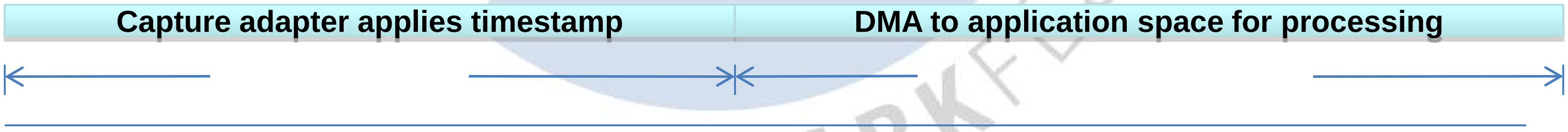
- Kernel copy of packet
- Inconsistent interrupt latency
- Application or capture library (libpcap) must perform time stamping
- System processing activity effects timestamp accuracy



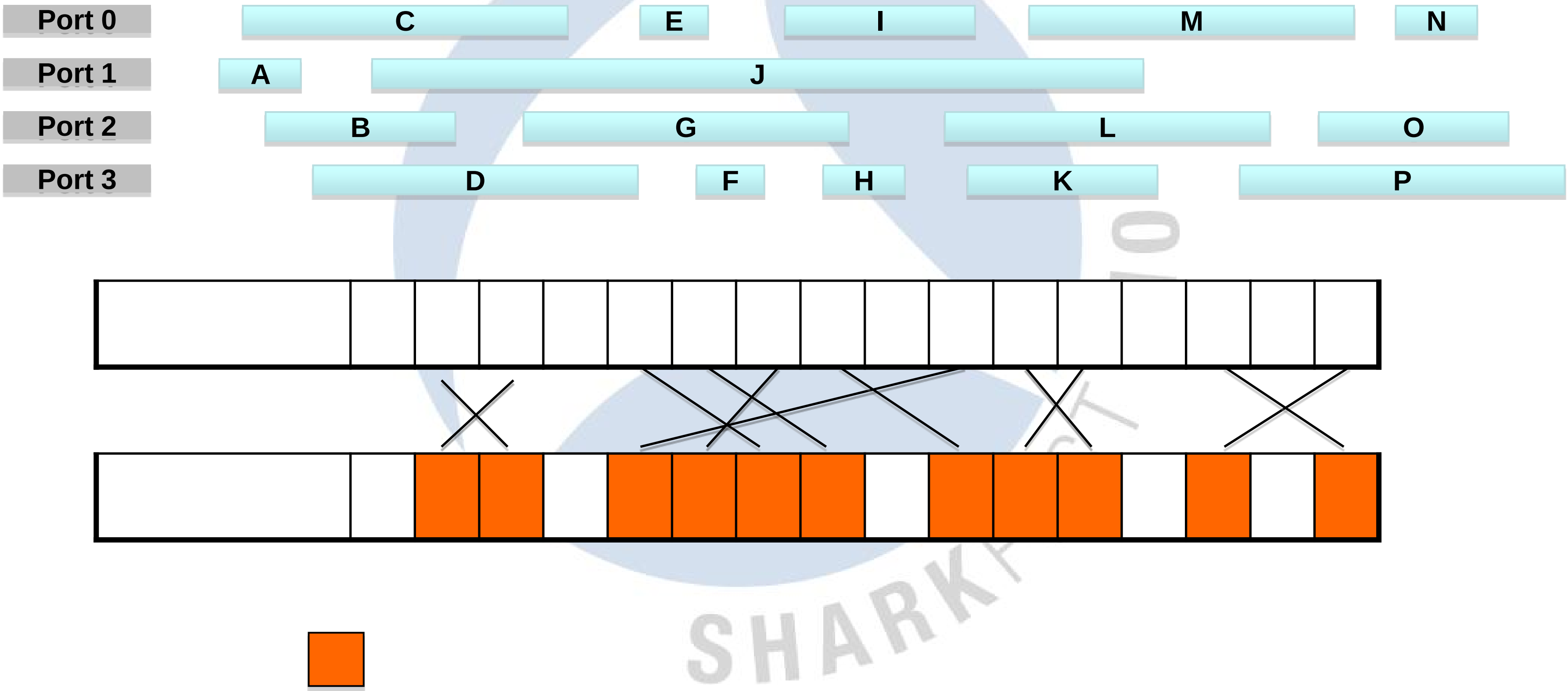
NIC | IRQ Latency | Kernel Copy | Software applied timestamp

Packet time stamping and precision

- Specialized network adapters provide consistent time stamping behavior
  - Timestamp applied by hardware consistently at the same position within the captured frame
  - Captured frame is delivered to application along with timestamp
- Because timestamp is applied by the hardware:
  - Interrupt latency and kernel copying have no affect on timestamp accuracy
  - System processing ambiguity has zero affect on timestamp accuracy


Capture adapter applies timestamp
DMA to application space for processing

Packet time stamping and precision

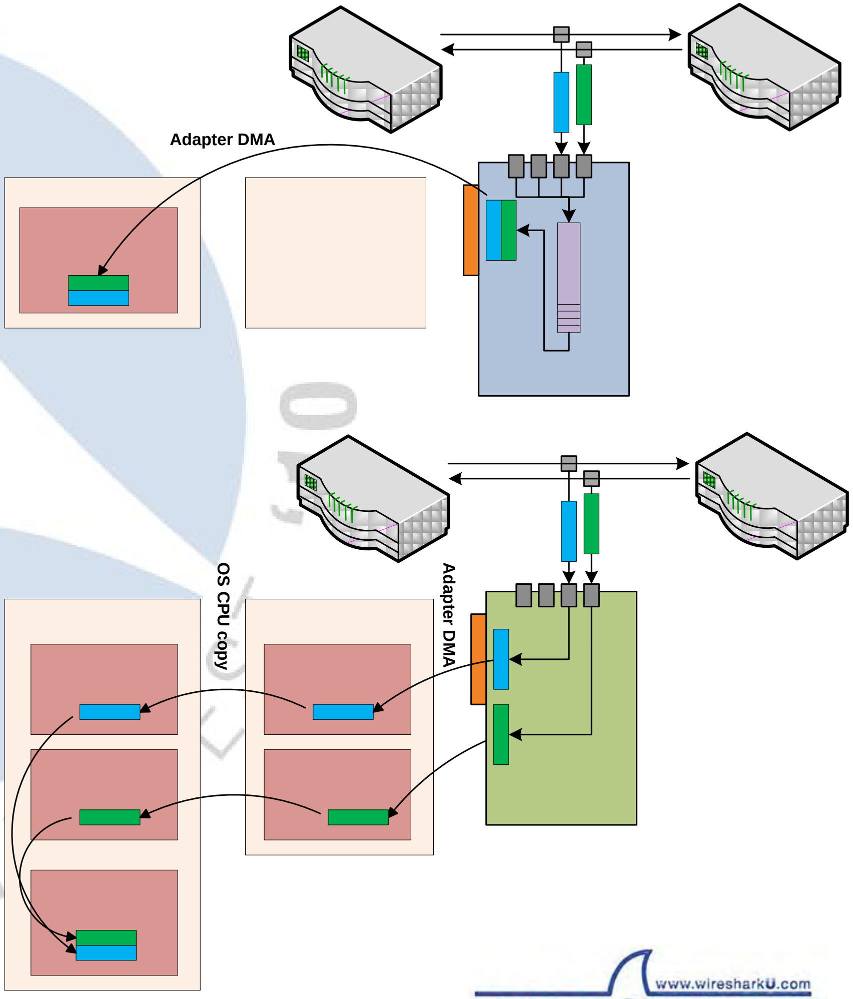


To overview

Packet time stamping and precision

All Napatech adapters support merging of streams.

- When applications need to process both RX and TX data from a link, it is often important to process the request-response traffic in the correct order.
- The Napatech adapters support merging of data from 2 or more ports into a stream.
- Merging of data is done based on the frame reception time. This means that request-response traffic will always be delivered to the host in the correct order.
- Processing of packets in time order can be important:
  - When data is to be analyzed on the fly.
  - When data is to be stored for later analysis.
- This functionality enables higher host processing performance.



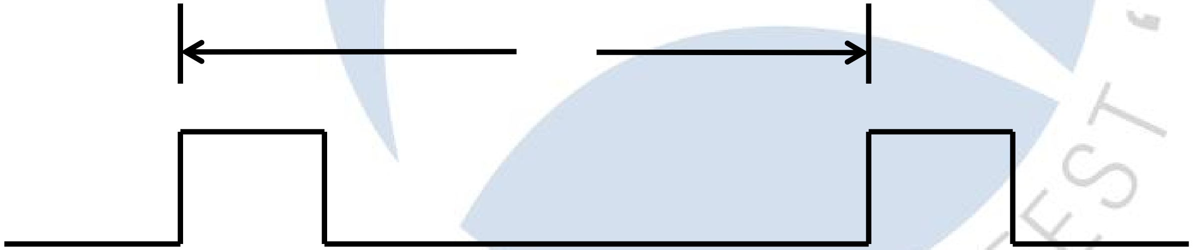
Standard NICs do not have this functionality, which means that received data must be sorted by the host CPU.

- Sorting frames in time order by the host CPU is CPU intensive and difficult.
- If data is to be stored on disk in time order, an extra CPU memory copy is needed.

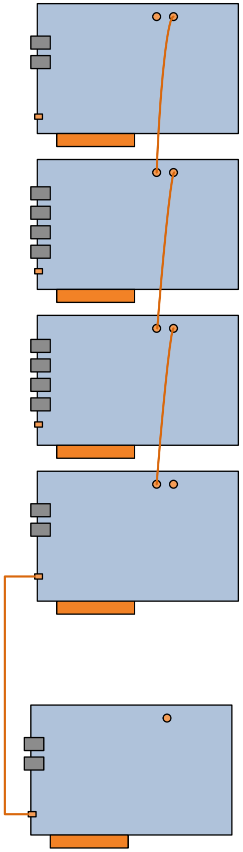
Packet time stamping and precision

Feature	Napatech Capture Adapter	Standard NIC
		› ›
		› › ›

To overview



To overview



Adapter	Adapter Time Stamping	Time Synchronization					Ethernet minimum frame time
		Adapter-to-adapter HW	Daisy-chain	Time Sync Unit	GPS time	OS	

Eliminating Packet Loss

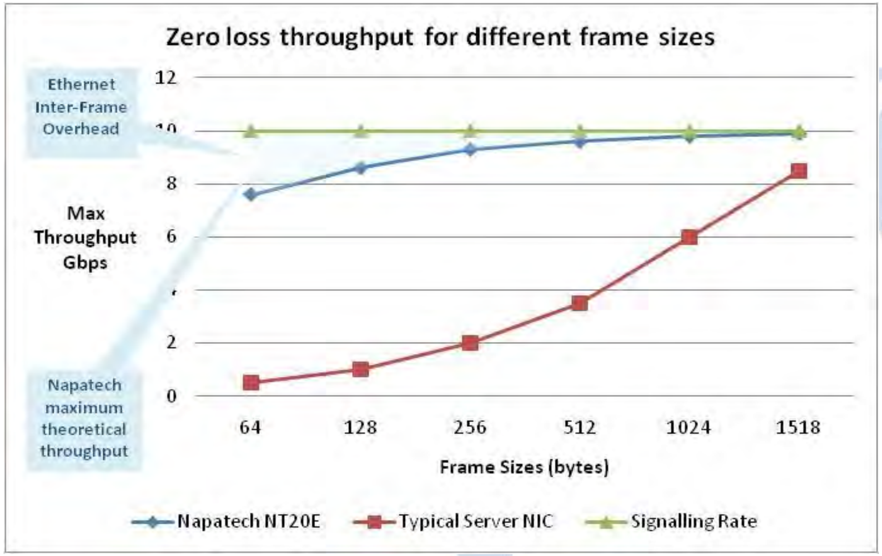
Standard NICs are built for efficient data communications.

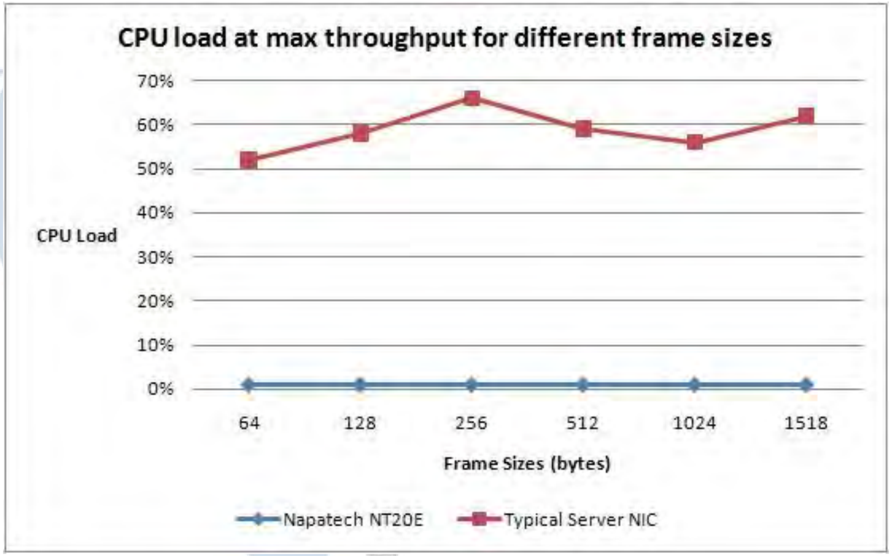
Napatech specialized adapters are built for efficient packet capture, analysis and transmit.

Napatech adapters differ by design:

	Napatech Adapters	Standard NICs

Eliminating Packet Loss





Napatech network adapters provide throughput performance equal to theoretical maximum. (Ethernet overhead reduces maximum throughput at smaller frame sizes).

Standard adapters can approach theoretical maximum performance for large frames, but are extremely poor at handling small frame sizes.

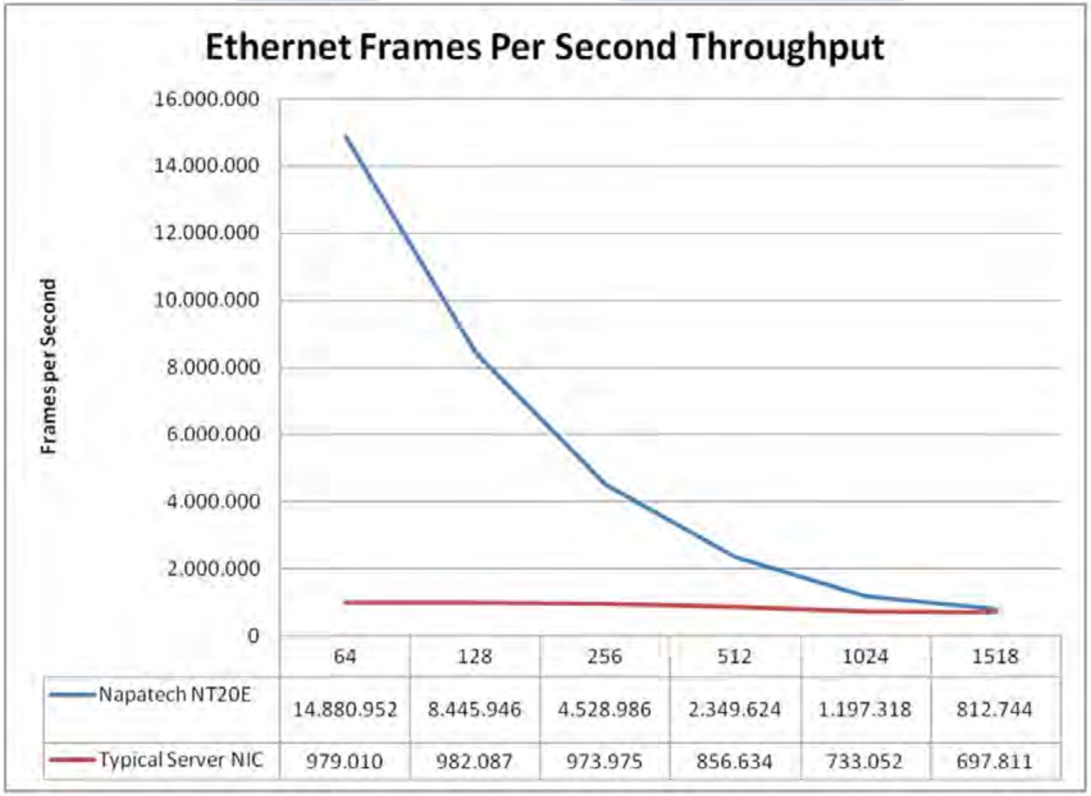
Standard adapters also use considerable CPU processing power to process frames.

Napatech Capture Adapters exhibit less than 1% load for all packet sizes.

Eliminating Packet Loss

Ethernet throughput can also be viewed in terms of packets per second or Ethernet frames per second

The distinction between typical server NICs and Napatech becomes clearer as Napatech network adapters are built for handling large numbers of frames



	64	128	256	512	1024	1518
Napatech NT20E	14.880.952	8.445.946	4.528.986	2.349.624	1.197.318	812.744
Typical Server NIC	979.010	982.087	973.975	856.634	733.052	697.811

Eliminating Packet Loss: Review

Napatech Adapter Feature	Benefit	Standard Network Adapters
Frame burst buffering on adapter		
Long PCI bursts		
Large host buffers		
OS bypass, zero copy of captured packets directly to user application memory		
Merging of streams		

Advanced Features: Filters

Filter functionality:

- 64 fully programmable filters.
- Received frames can be filtered at full line speed for all frame sizes and all combinations of filter settings.
- Dynamic offset of filters, based on automatic detection of packet type:
  - 26 predefined fields (Ethernet, IPv4, IPv6, UDP, TCP, ICMP, …) (see next slide for example)
- Fixed offset relative to dynamic offset position.
- Predefined filters: IPv6, IPv4, VLAN, IP, MPLS, IPX.
- 64-byte patterns can be matched.
- The length of the received frame can be used for filtering frames.
- The port on which the frame was received can be used for filtering.

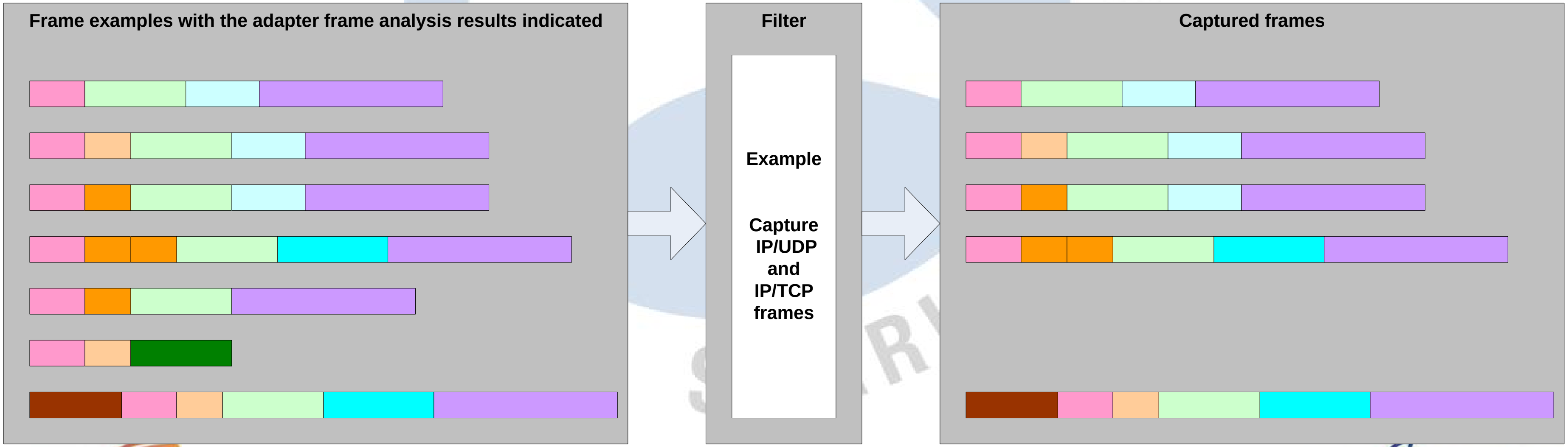
Benefits:

- Enables filtering of network frames so that the user application only needs to handle relevant frames, off-loading the user application.
- Filtering can be done at network line speed.

Advanced Features: Filters

- Filter Example
  - The figure below illustrates how filters can be used to capture IP/UDP and IP/TCP frames
  - NTPL syntax:

```
 Capture[Priority=0; Feed=0] = ((Layer3Protocol == IP) AND
        ((Layer4Protocol == UDP) OR (Layer4Protocol == TCP)))
```


Frame examples with the adapter frame analysis results indicated
Filter
Example
Capture IP/UDP and IP/TCP frames
Captured frames

Advanced Features: Fixed Slicing (snaplen)

All Napatech adapters support fixed slicing.

Using fixed slicing it is possible to slice captured frames to a fixed maximum length before they are transferred to the user application memory.

Libpcap snaplen translated to hardware slicing so no additional programming is required

The fixed slicing can be configured using the NTPL:

Example: Slice captured frames to a maximum length of 128 bytes:

```
Slice[Priority=0; Offset=128]  = all
```

Advanced Features: Multi CPU Buffer Splitting

Multi CPU buffer splitting enables the adapter to distribute the processing of captured frames among the host CPUs.

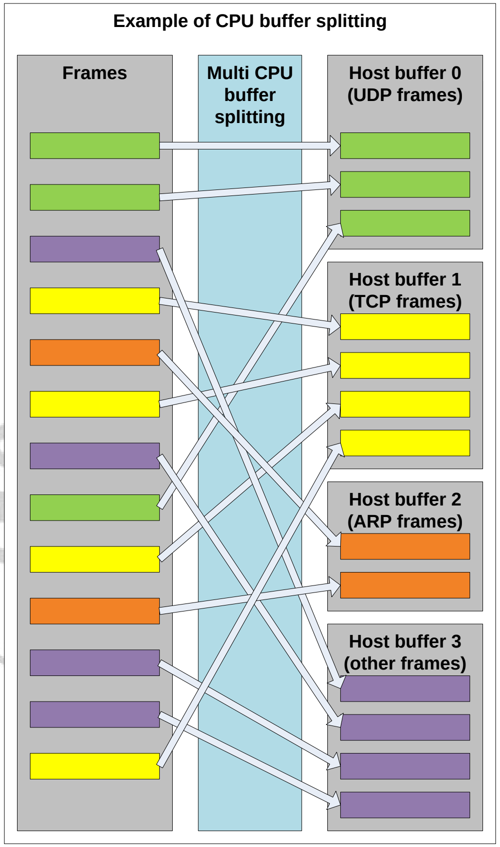
- CPU load distribution is hardware-accelerated.
- Captured data is placed in separate buffers for the different CPU cores in the host system.

The multi CPU buffer splitting functionality can be configured to place data in 1, 2, 4, 8, 16 or 32 different host buffers.

- The algorithm used by the adapter for placing a captured frame in a host buffer is based on packet flow information and or protocol filter.

Flows can be defined by:

- The results from the filter logic including port numbers
- The generated hash key value
- A combination of the above 2 possibilities


Example of CPU buffer splitting
Frames
Multi CPU buffer splitting
Host buffer 0 (UDP frames)
Host buffer 1 (TCP frames)
Host buffer 2 (ARP frames)
Host buffer 3 (other frames)

Advanced Features: Multi CPU Buffer Splitting

NTPL is used to define multi CPU host buffer splitting.

Example 1 (using the filter logic, see also the figure at previous slide):

```
HashMode = None
Capture[Priority=0; Feed=0] = (Layer4Protocol == UDP)
Capture[Priority=0; Feed=1] = (Layer4Protocol == TCP)
Capture[Priority=0; Feed=2] = (Layer3Protocol == ARP)
Capture[Priority=0; Feed=3] = (((Layer4Protocol !== UDP) AND (Layer4Protocol != TCP)) AND
                               (Layer3Protocol != ARP))
```

Example 2 (using 5-tuple hash, data distributed to 16 host queues):

```
HashMode = Hash5Tuple
Capture[Priority=0; Feed=(0..15)] = All
```

Example 3 (using a combination of filter logic and hash key to define flows):

```
HashMode = Hash5TupleSorted
Capture[Priority=0; Feed=(0..3)] = (mUdpSrcPort == (16000..16500))
Capture[Priority=0; Feed=4,5]    = (mTcpSrcPort == mTcpPort_HTTP)
Capture[Priority=0; Feed=6]      = (((Layer3Protocol == IP) AND
                                     (mUdpSrcPort != (16000..16500))) AND
                                    (mTcpSrcPort != mTcpPort_HTTP))
Capture[Priority=0; Feed=7]      = (mMacTypeLength == mMacTypeLength_ARP)
```

Napatech LibPCAP Library

The current Napatech LipPCAP is based on the LipPCAP 0.9.8_2.1.A release

Delivered as open source ready to configure and compile.

Linux and FreeBSD supported

Support for all feed configurations supported by the NT adapters:

- Packet feeds can be configured at driver load time
- Feeds are configured using simple NTPL syntax.
- Feeds are started and stopped through libpcap

Full support for protocol filters configuration via NTPL scripts.

snaplen (-s option in tshark) translated to slicing in hardware

Napatech LibPCAP Library

Example. Build and install of new LibPCAP:

Extract standard LibPCAP distribution:

```
# tar xfz napatech_libpcap_0.9.8-x.y.z.tar.gz
```

Configure LibPCAP:

```
# autoconf
# ./configure --prefix=/opt/napatech  --with napatech=/opt/napatech
```

Build the shared library version of LibPCAP:

```
# make shared
```

As root, install the shared library:

```
# make install-shared
```

Simple Wireshark installation:

```
# ./configure –with-libpcap=/opt/napatech
# make
# make install
```

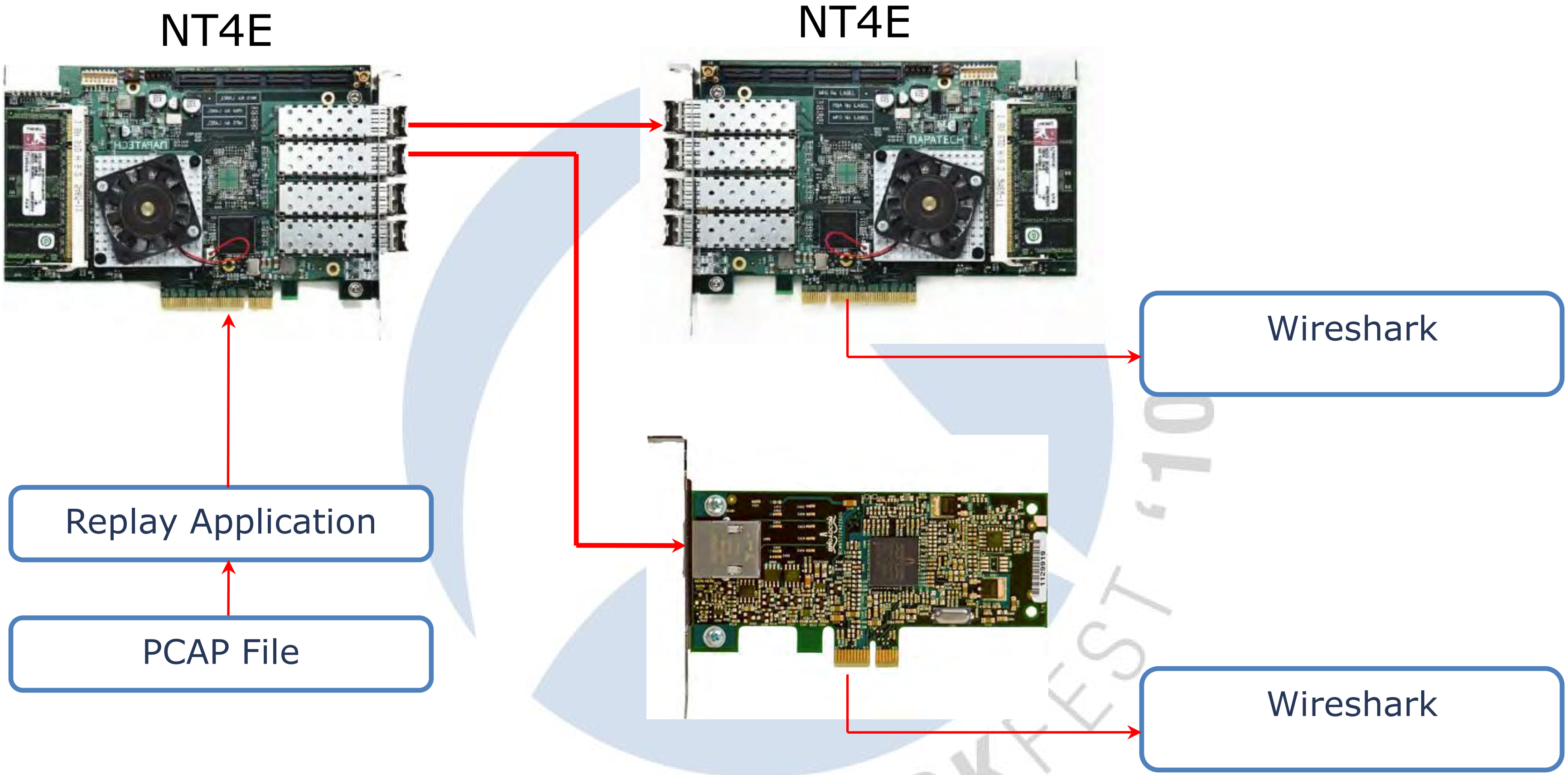
Napatech LibPCAP Library

Configuration example showing how to setup adapter to capture HTTP frames and distribute them to 8 host buffers using a 5-tuple hash key.

```
DeleteFilter = All
SetupPacketFeedEngine[ TimeStampFormat=PCAP; DescriptorType=PCAP;
    MaxLatency=1000; SegmentSize=4096; Numfeeds=8 ]
PacketFeedCreate[ NumSegments=128; Feed=(0..6) ]
PacketFeedCreate[ NumSegments=16; Feed=7 ]
HashMode = Hash5TupleSorted
Capture[ Feed = 0..6 ] = mTcpSrcPort == mTcpPort_HTTP
Capture[ Feed = 7 ] = Layer3Protocol == ARP
```

Eight LipPCAP applications can be started to handle frames from the "ntxc0:0","ntxc0:1", "ntxc0:2", … "ntxc0:7" virtual adapter devices.

Sharkfest 2010 Demonstration



NT4E

NT4E

Wireshark

Replay Application

PCAP File

Wireshark

About Napatech

Napatech is a leading OEM supplier of the highest performing 1 & 10 Gb/s Hardware Acceleration Network Adaptors

Application offloading through hardware acceleration:

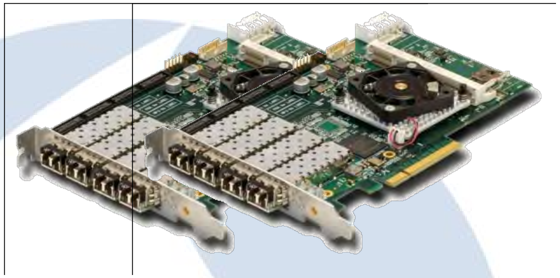
- A flexible Feature-Upgradable FPGA technology
- A scalable migration path from 1 Gb/s to 10 Gb/s networks, and beyond
- A Uniform platform API that is easy to integrate and maintain
- Industry standard LibPCAP support

Denmark
Copenhagen
Admin

USA East Coast
Boston, MA

USA West Coast
Mountain View, CA

NT4E-STD Adapter



NT4E + NTPORT4

NT20E Adapter