

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

June 17th 2010

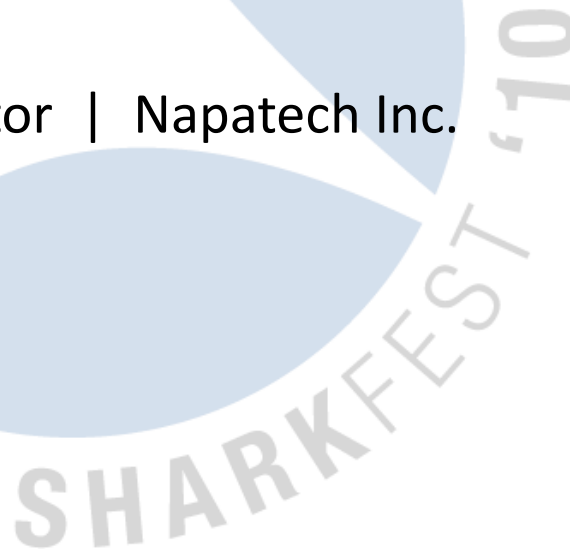
Pete Sanders

Application Engineering Director | Napatech Inc.

SHARKFEST '10

Stanford University

June 14-17, 2010



Agenda

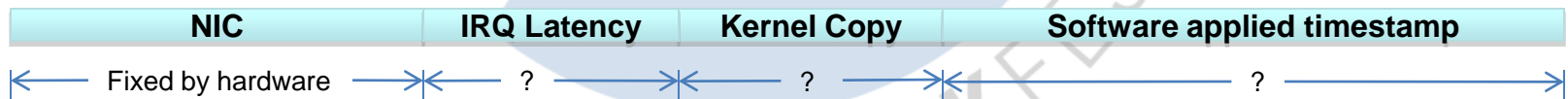
- Network analysis minimum requirements
- Packet time stamping and precision
- External time synchronization
- Eliminating packet loss
- Advanced features of Napatech network adapters
- Napatech libpcap support
- Demonstration

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

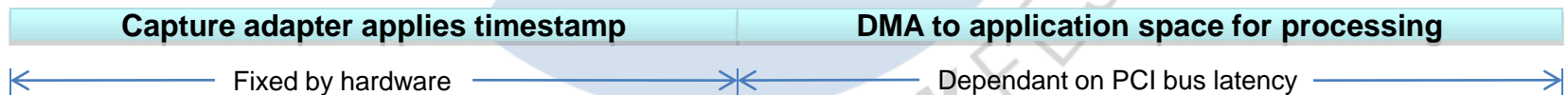
- Standard network interfaces provide non-deterministic time stamping behavior
 - Kernel copy of packet
 - Inconsistent interrupt latency
 - Application or capture library (libpcap) must perform time stamping
 - System processing activity effects timestamp accuracy



Timestamp processing timeline for standard NIC

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 effect on timestamp accuracy
 - System processing ambiguity has zero effect on timestamp accuracy



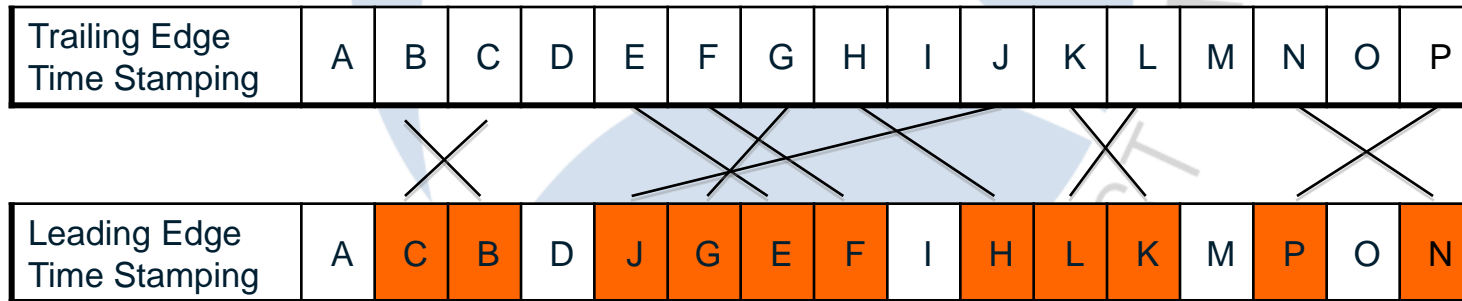
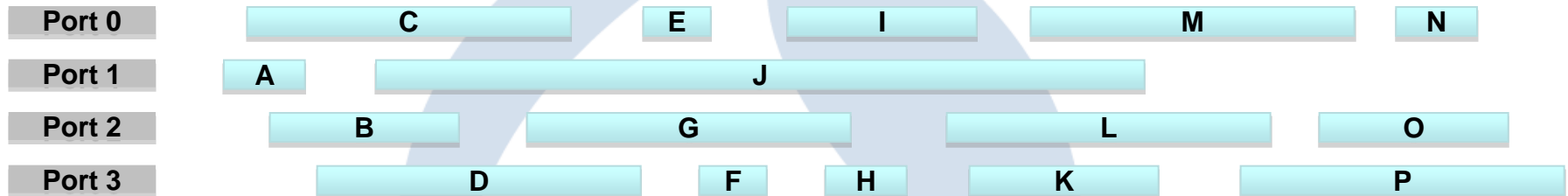
Timestamp processing timeline for specialized capture adapter

Packet time stamping and precision

- High precision hardware based time stamping enables:
 - High precision latency measurements and optimization.
 - Network performance analysis and optimization.
- Implementing time stamping in software provides very low precision.
 - Software time stamping is typically 1000 times less accurate than hardware based time stamping.
- Hardware generation of multiple time formats enabled the application to receive the time stamp format best matched to the application.
 - Converting time stamp formats in software is CPU intensive.
- Time stamping frames when the last byte of the frame is received.
 - Time stamping at the end of the frames ensure that frames received on multiple ports always are time stamped in the order captured.

Packet time stamping and precision

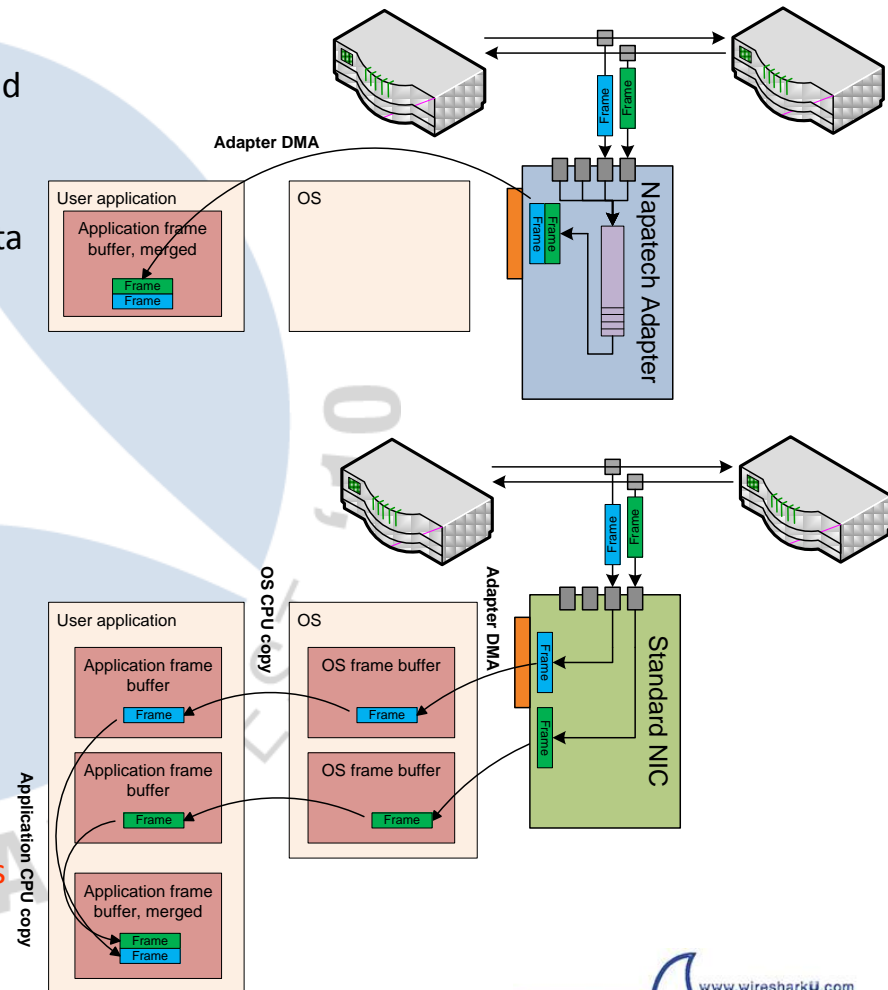
> 4-port time stamping example.



 = Packet time stamp order different from capture order

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

Specialized vs. Standard NIC Comparison

Feature	Napatech Capture Adapter	Standard NIC
Inter frame precision (one port)	< 10 ns \pm 10 ns. Typically 6.7 ns.	< 1,000,000 ns. > 10 G : 800-28000 minimum frames. > 1 G: 80-2800 minimum frames.
Frame precision between ports	< 10 ns \pm 10 ns. Typically 6.7 ns.	< 1,000,000 ns. > 10 G: > 1000 minimum frames. > 1 G: > 100 minimum frames. > The order of frames cannot be determined.
Performance	All time-stamping is done by the adapter hardware. No CPU processing is needed.	Must be done by software. Can be time-consuming when many frames are received.
Time stamp formats	6 time stamp formats are supported including PCAP micro and nano time	Converting between different time formats in real time wastes CPU cycles.

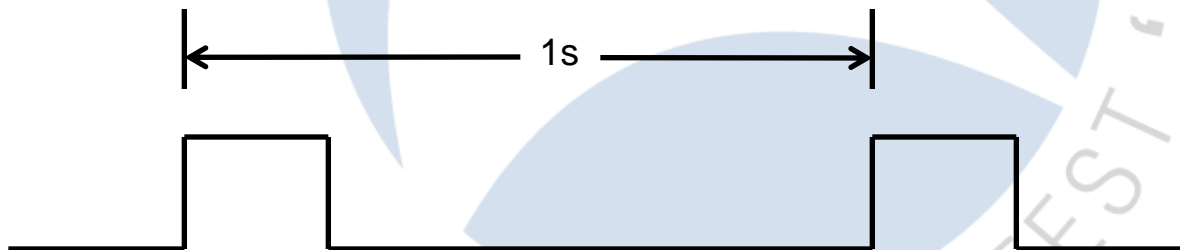
External Time Synchronization: GPS

- Geo-site absolute latency measurements
- Synchronized to $< 150\text{ns}$ of GPS/UTC
- All time stamp formats supported
- Synchronize multiple adapters/appliances
- Very high inter-adapter time-stamping precision. $< 30\text{ ns}$.
Typical 7 ns .



External Time Synchronization: PPS

- Napatech adapters support sampling of a 1 second clock pulse (PPS) received from an external source.
- This functionality enables the application make precise relative packet time measurements synchronized to an external source
- One second UTC time from external source can be appended for absolute measurements

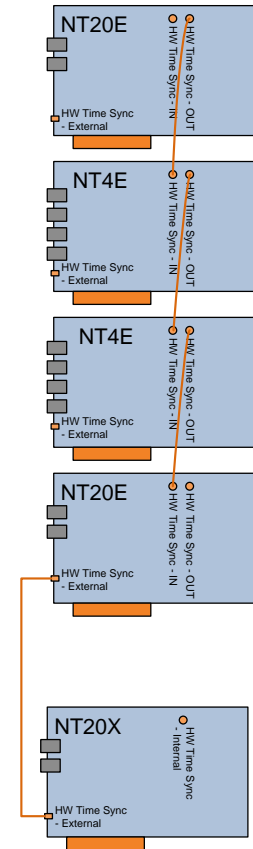


External Time Synchronization: OS Time

- All Napatech adapters support OS time synchronization.
- OS time synchronization enables the adapter time (the time used by the adapter to time stamp frames) to be synchronized with the time of the operating system.
- The adapter HW precisely synchronizes to the OS time:
 - The adapter time is changed in small steps so that the inter-frame time stamp will never perform large jumps
 - The adapter time and the operating system time are synchronized every 200 ms by the adapter driver.
- OS time synchronization can be used when very high time synchronization is not required:
 - The host server can get the time via NTP, 1588, etc.
- The precision of OS time synchronization is a factor 20-200 lower than hardware time synchronization.

Adapter-to-adapter HW Time Synchronization

- The Napatech adapters support hardware time synchronization of time stamping between multiple adapters.
- The solution is high precision and cost effective solution for synchronization of multiple adapter using a small coax cable.
- Napatech adapters support daisy chain time synchronization of multiple adapters (see the figure at the right).



External Time Synchronization: Summary

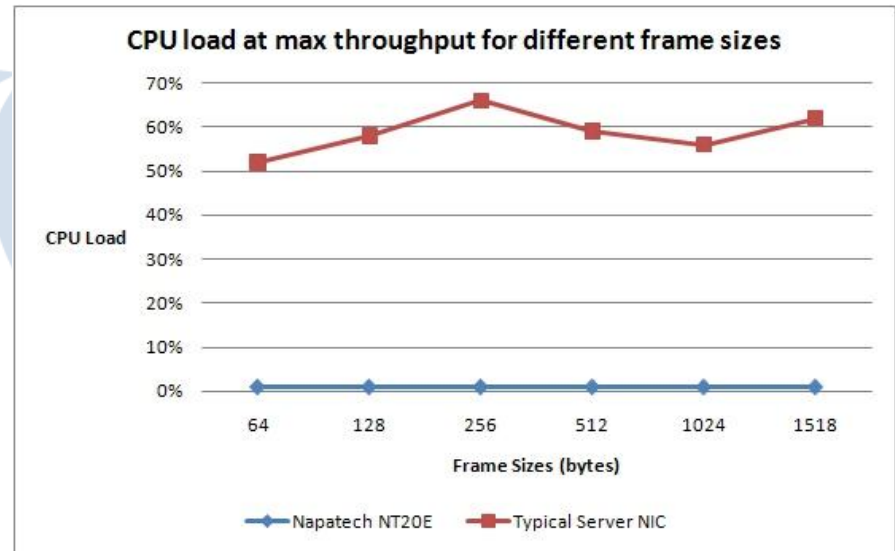
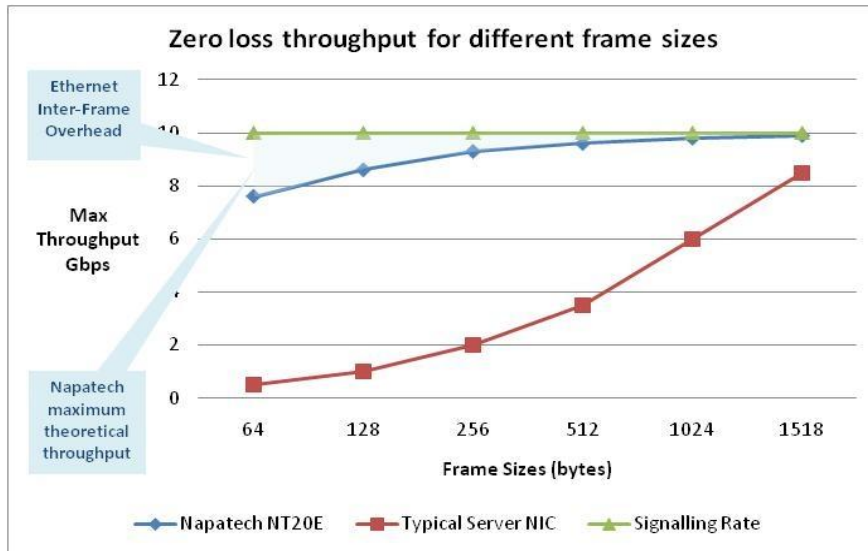
Adapter	Adapter Time Stamping	Time Synchronization					Ethernet minimum frame time
		Adapter-to-adapter HW	Daisy-chain	Time Sync Unit	GPS time	OS	
NT20E (10Gbps)	Accuracy < 20 ns.	Accuracy < 30 ns.	Accuracy < 30 ns.	Accuracy < 30 ns.	Accuracy < 150 ns.	Typical < 300 ns.	70 ns
NT4E (1Gbps)	Typical 6.7 ns.			Typical 7 ns.	(typical 30 ns).		704 ns

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
Design criteria	High-volume capture, analysis and transmit processing	High-speed point-to-point communication
Packet transfer to host	Burst-packet transfer	Single-packet transfer
Burst protection	On-board memory	Small amount
Packet transfer to application	DMA zero-copy	Interrupt driven OS copy
CPU load	<1% across all frame sizes and packet rates	Governed by packet rate and CPU performance
Packet acceleration features	Many	Few

Eliminating Packet Loss

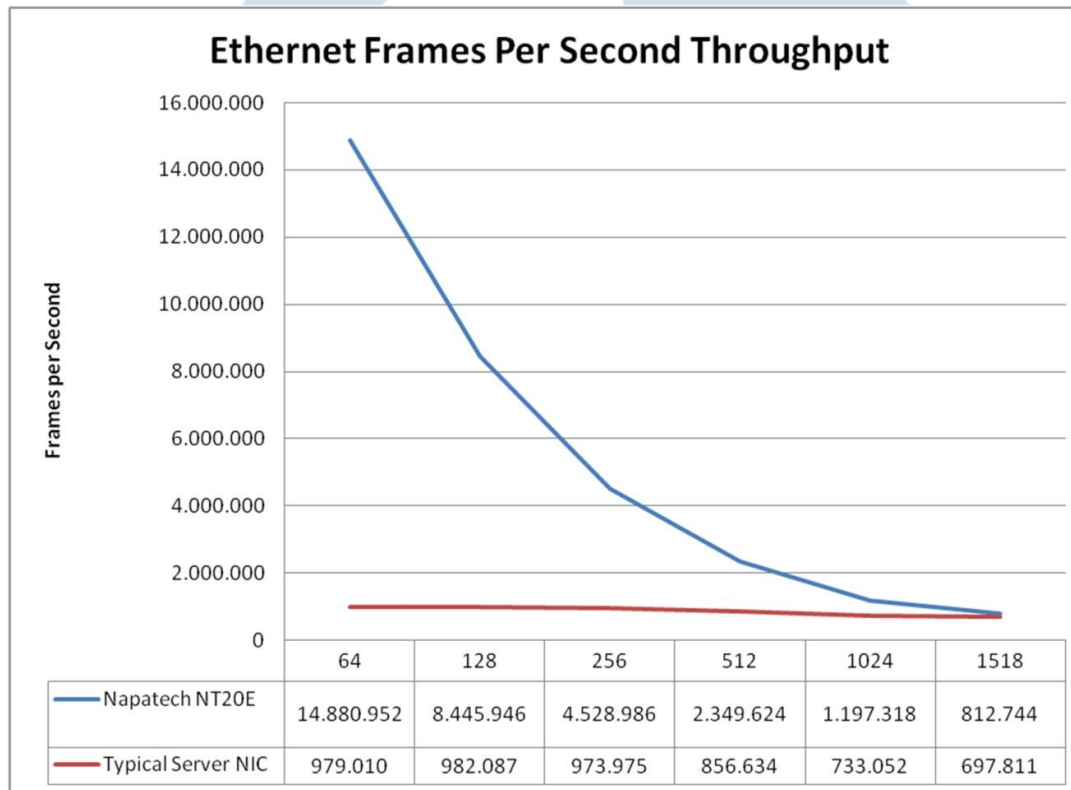


Source: CESNET

- 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



Eliminating Packet Loss: Review

Napatech Adapter Feature	Benefit	Standard Network Adapters
Frame burst buffering on adapter	No data is lost, even when captured data bursts exceed the PCI interface speed, or the PCI interface is temporarily blocked. For NT20X 2 x 10 Gbps can be handled down to 150 bytes frames. For NT20X 1 x 10 Gbps can be handled at any frame size.	Data is lost, when captured bursts exceed the PCI interface speed, or the PCI interface is temporarily blocked.
Long PCI bursts	Very high PCI performance can be achieved for all frame sizes.	The PCI performance will depend on the frame size. E.g. the overhead for a 64-byte frame can be as much as 45%, while for a 1-KB frame it will be only 5%.
Large host buffers	Data can be processed at much higher speed, and the frame processing overhead is much lower (releasing processing power to the user application).	Frames are handled one at a time giving a large processing overhead resulting in lower user application speed.
OS bypass, zero copy of captured packets directly to user application memory	There is no packet copying or OS handling overhead.	A standard OS packet handling interface performs one or more copy of all frames resulting in lower application speed.
Merging of streams	Adapters can merge packets received on 2 or more ports in reception time order, whereby the host CPU is off-loaded.	The sorting of frames in time order must be done by the host CPU, reducing the possible host processing performance.

Advanced Features: Filters

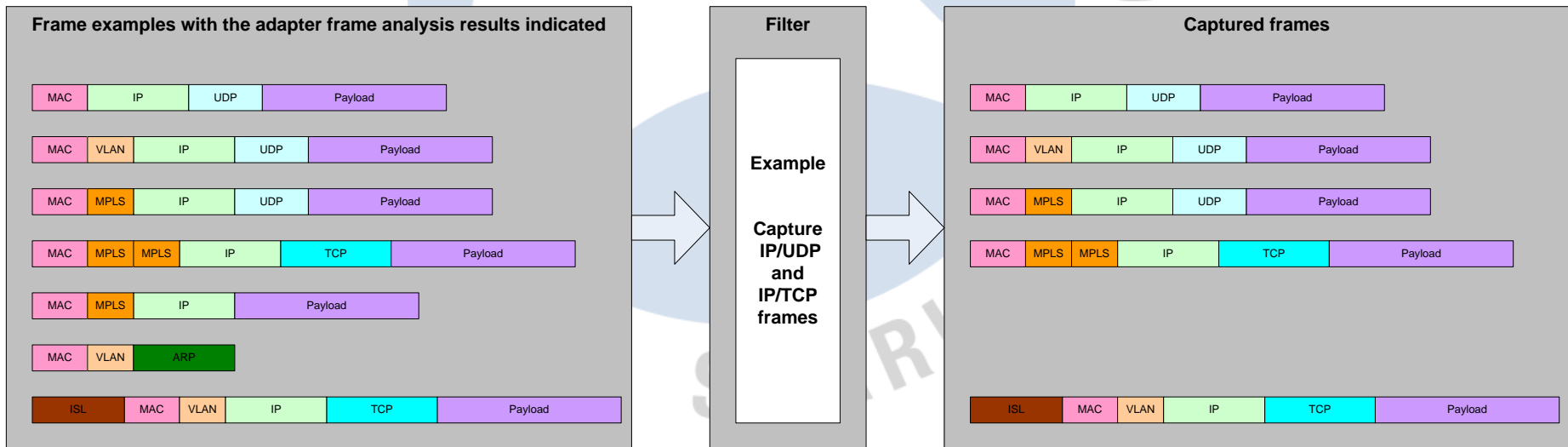
- Filter functionality:
 - 64 fully programmable filters.
 - Received frames can be filtered at full line speed for all frame sizes and all combinations of filter settings.
 - Dynamic offset of filters, based on automatic detection of packet type:
 - 26 predefined fields (Ethernet, IPv4, IPv6, UDP, TCP, 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
- NTPS syntax:

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

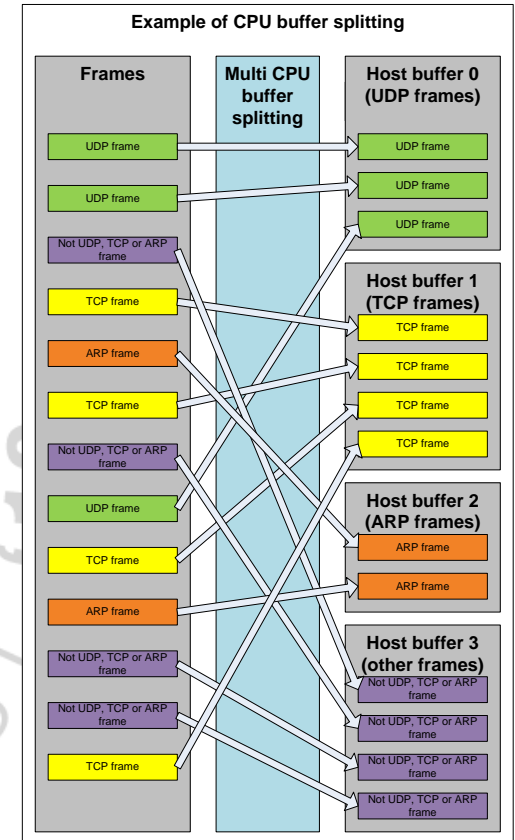


Advanced Features: Fixed Slicing (snaplen)

- 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



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 xzf 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 ]
```

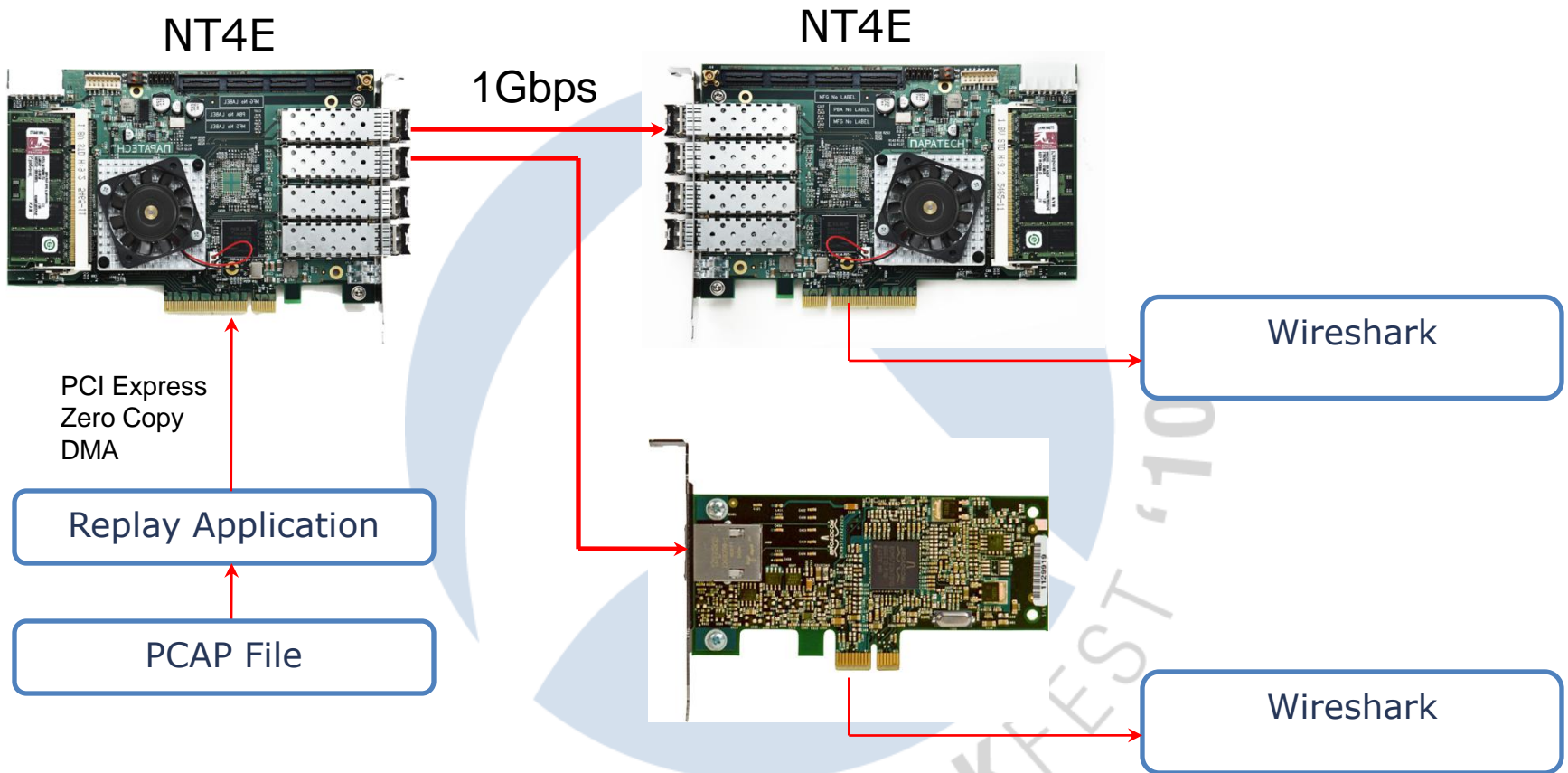
```
HashMode = Hash5TupleSorted
```

```
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



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
HQ, R&D and Admin



USA East Coast
Boston, MA
Sales & Support



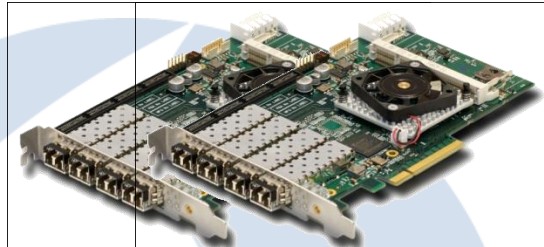
USA West Coast
Mountain View, CA
Sales & Support

Napatech Adapter Portfolio – PCIe



NT4E-STD Adapter

- 4x1Gb Ethernet interface SFP or RJ45
- ½ Length PCI-E
- 4 Gbps lines speed capture
- Time stamping
- Host OS time sync
- Host-based retransmit
- CPU utilization: <1%
- Linux, FreeBSD and Windows drivers
- 3 different product variants available



NT4E + NTPORT4

- 4x1Gb Ethernet interface SFP or RJ45
- ½ Length PCI-E
- External time sync connector
- 8Gbps packet processing, filtering, tagging, timestamp, slicing, local retransmit
- CPU utilization: <1%
- Linux, FreeBSD and Windows drivers
- 3 different product variants available



NT20E Adapter

- 2x10Gb Ethernet interface XFP
- ½ Length PCI-E
- External time sync connector
- 20G packet processing, filtering, tagging, timestamp, slicing, local retransmit
- CPU utilization: <1%
- Linux, FreeBSD, and Windows drivers