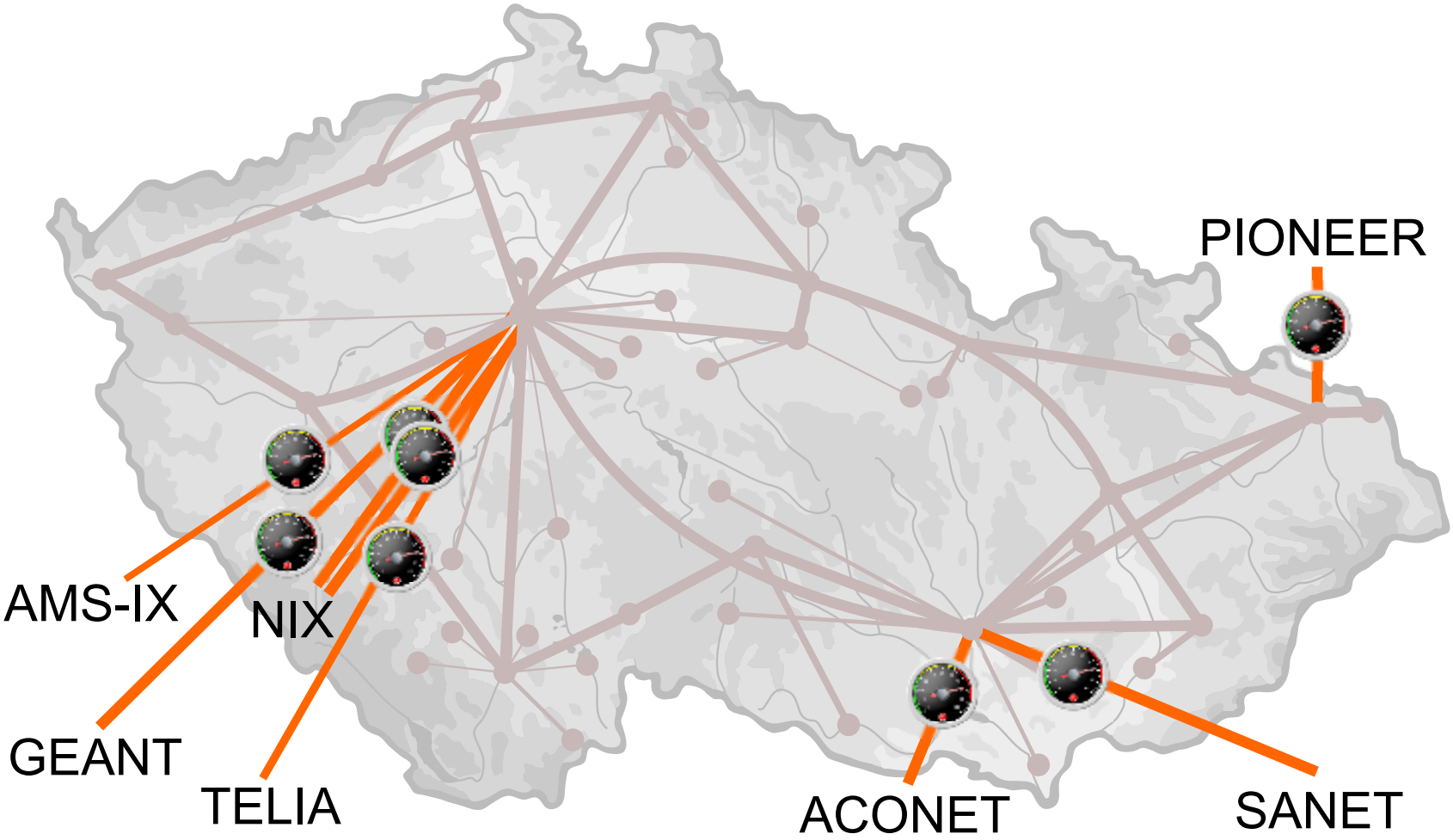
Software Defined Monitoring: Research Platform for High Speed Network Monitoring

(31st NMRG Meeting – Zürich, Switzerland)

Lukáš Kekely, Viktor Puš, Jan Kořenek
(kekely,pus,korenek@cesnet.cz)

14. 10. 2013

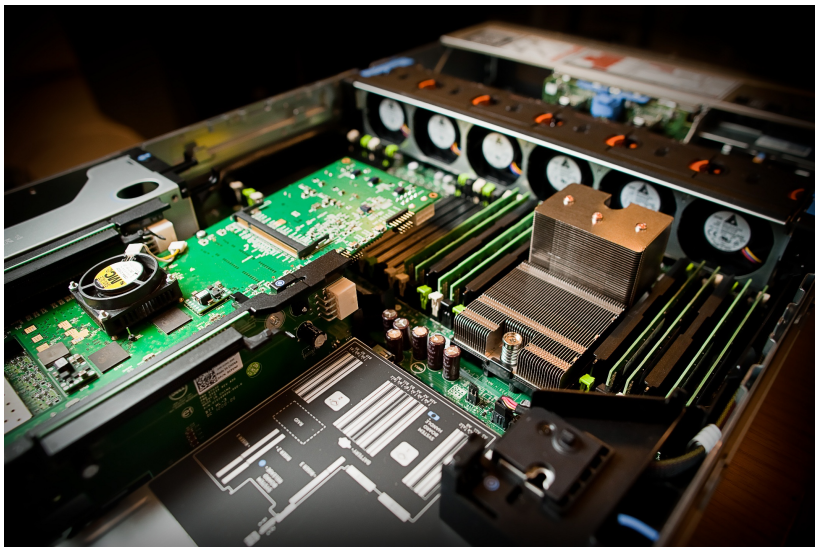
Czech NREN Cesnet



- metering points on the edges of the network (highlighted)

Current Metering Point

- commodity server running Linux
- SW flow exporter (NetFlow/IPFIX) from SME Invea-Tech
  - support for creation of traffic processing plugins
- our own hardware probe from COMBOv2 family
  - PCI-Express card with two 10 GbE ports and Virtex5 FPGA
  - HaNic over NetCope as firmware – packet capture, precise timestamps (ns), flow based traffic division ...



Motivation

$\Rightarrow$ **We want more than that!**

$\Rightarrow$ **We want more than that!**

1. **Higher speed**
   - constant advances in the network bandwidth
   - monitored links are going to be upgraded to 40/100 Gbps
2. **Higher quality**
   - more than just classical NetFlow statistics
   - flexible additional data according to actual need
   - application protocol parsing and deep packet inspection

$\Rightarrow$ **We want more than that!**

1. **Higher speed**
   - constant advances in the network bandwidth
   - monitored links are going to be upgraded to 40/100 Gbps
2. **Higher quality**
   - more than just classical NetFlow statistics
   - flexible additional data according to actual need
   - application protocol parsing and deep packet inspection

Problem: Current CPUs are not fast enough to process whole traffic all alone!

$\Rightarrow$ **We want more than that!**

1. **Higher speed**
   - constant advances in the network bandwidth
   - monitored links are going to be upgraded to 40/100 Gbps
2. **Higher quality**
   - more than just classical NetFlow statistics
   - flexible additional data according to actual need
   - application protocol parsing and deep packet inspection

Problem: Current CPUs are not fast enough to process whole traffic all alone!
Solution: We created new approach to monitoring acceleration called Software Defined Monitoring!

Software Defined Monitoring

What is it?

- new approach to acceleration of network monitoring
- brings HW accelerated, application controlled reduction of traffic (packet processing offload)
- still performs packet capture, precise timestamps, flow based traffic division

What does it do?

- **Hardware** provides various methods of packet preprocessing and aggregation – **The Muscles**
- **Software** controls the actual usage of preprocessing on flow basis – **The Controller**
- **User applications** request the acceleration and perform advanced monitoring tasks – **The Intelligence**

Software Defined Monitoring

What is it?

- new approach to acceleration of network monitoring
- brings HW accelerated, application controlled reduction of traffic (packet processing offload)
- still performs packet capture, precise timestamps, flow based traffic division

What does it do?

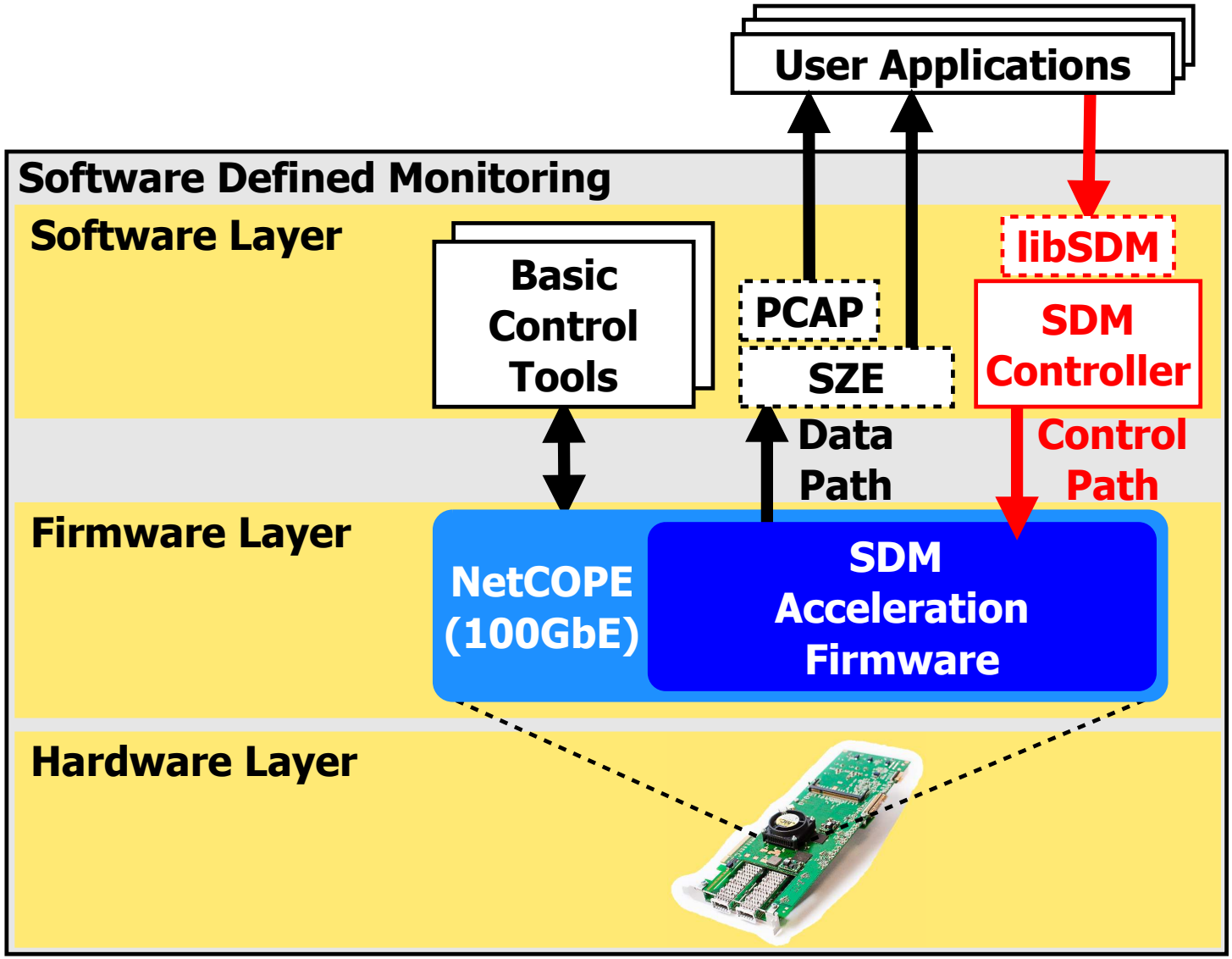
- **Hardware** provides various methods of packet preprocessing and aggregation – **The Muscles**
- **Software** controls the actual usage of preprocessing on flow basis – **The Controller**
- **User applications** request the acceleration and perform advanced monitoring tasks – **The Intelligence**

Applications adjust acceleration of traffic processing according to their actual needs!

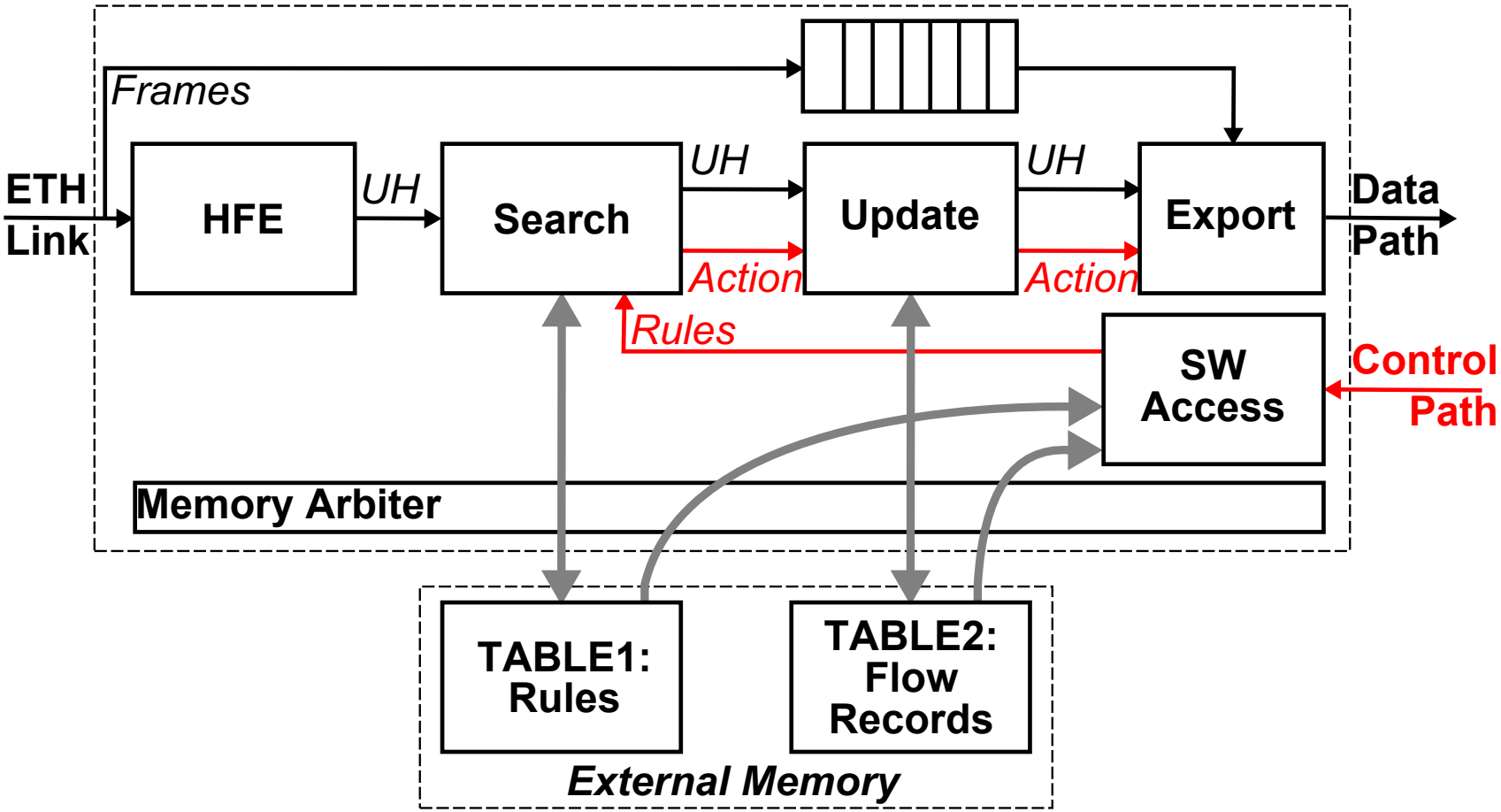
Traffic Preprocessing in Hardware

- fully controlled by rules from software
- four basic methods of frames preprocessing:
  - **Send** – preserve the whole frame (with payload)
  - **Extract** – preserve only basic data about the frame
  - **Aggregate** – update selected flow (NetFlow) record maintained in HW memory
  - **Drop** – simply ignore the frame

SDM Layered Scheme



SDM Firmware



SDM Use Cases

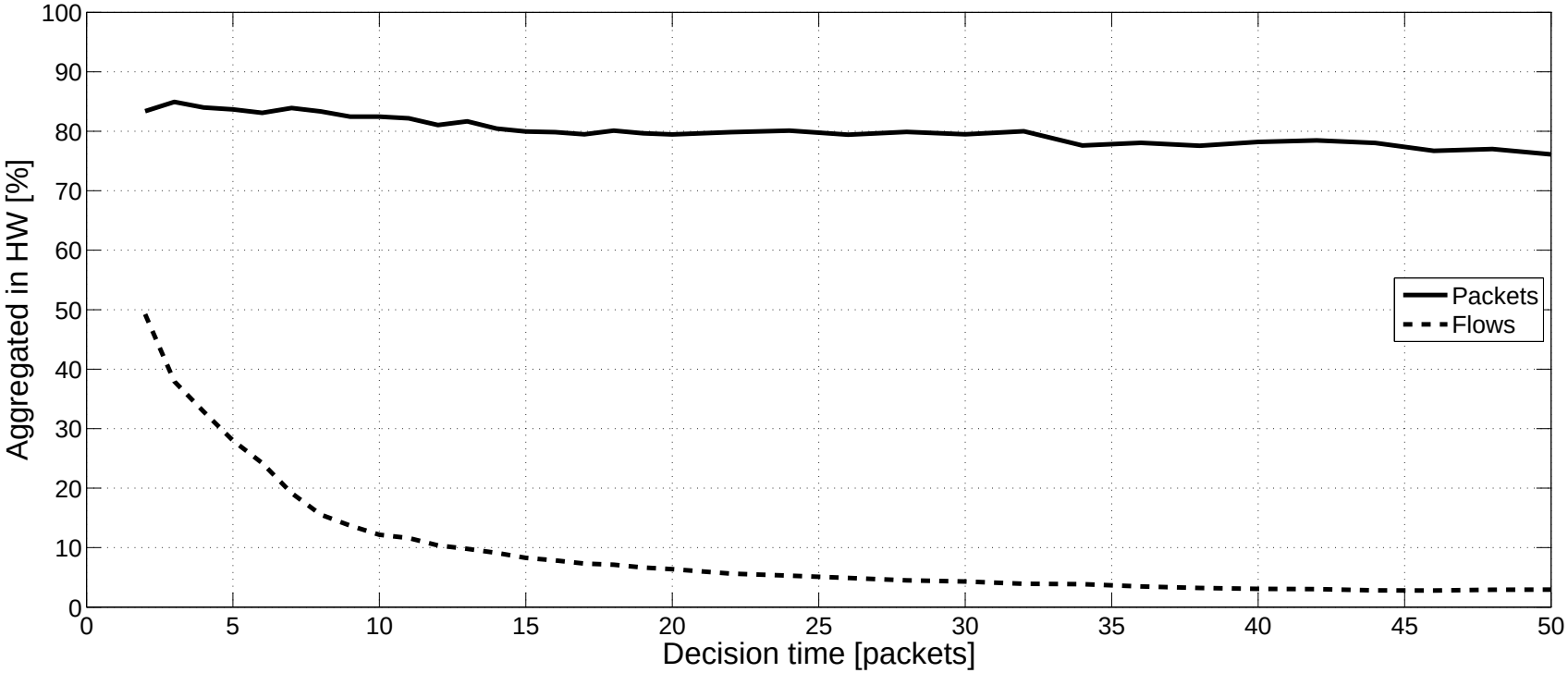
1. Basic NetFlow statistics
2. Application protocol parsing
3. Specific Non-NetFlow statistics
4. Lawfull interceptions
5. Forensic analysis of network traffic
   - "zoom-in" on suspicious data
6. Active SW networking device
   - accelerated switch, firewall, router . . .

7. Acceleration of your research application?

Basic NetFlow statistics

- useless payload of frames, but must have information about all incoming frames
  - **default:** use *Extract* on all traffic
  - **rules:** use *Aggregate* for selected (the heaviest) flows
- CPU performance savings:
  - no packet parsing at all
  - NetFlow aggregation computed only partially
- need to decide when to use NetFlow in HW based on the first X packets of flows

UC1: Results



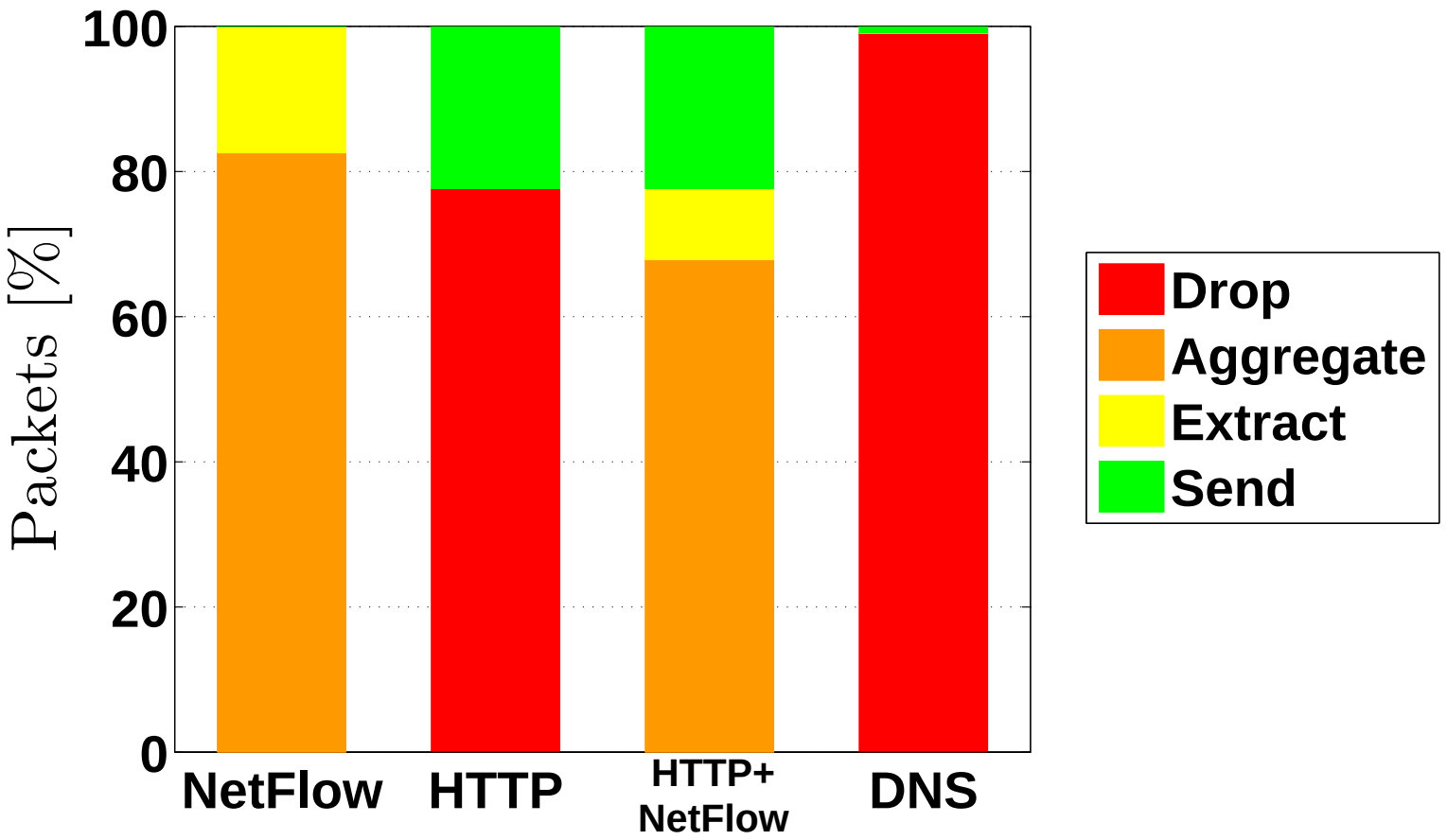
- the number of frames reduced to $\frac{1}{5}$ and data load to $\frac{1}{100}$

UC2: Details of SDM Usage

Application protocol parsing

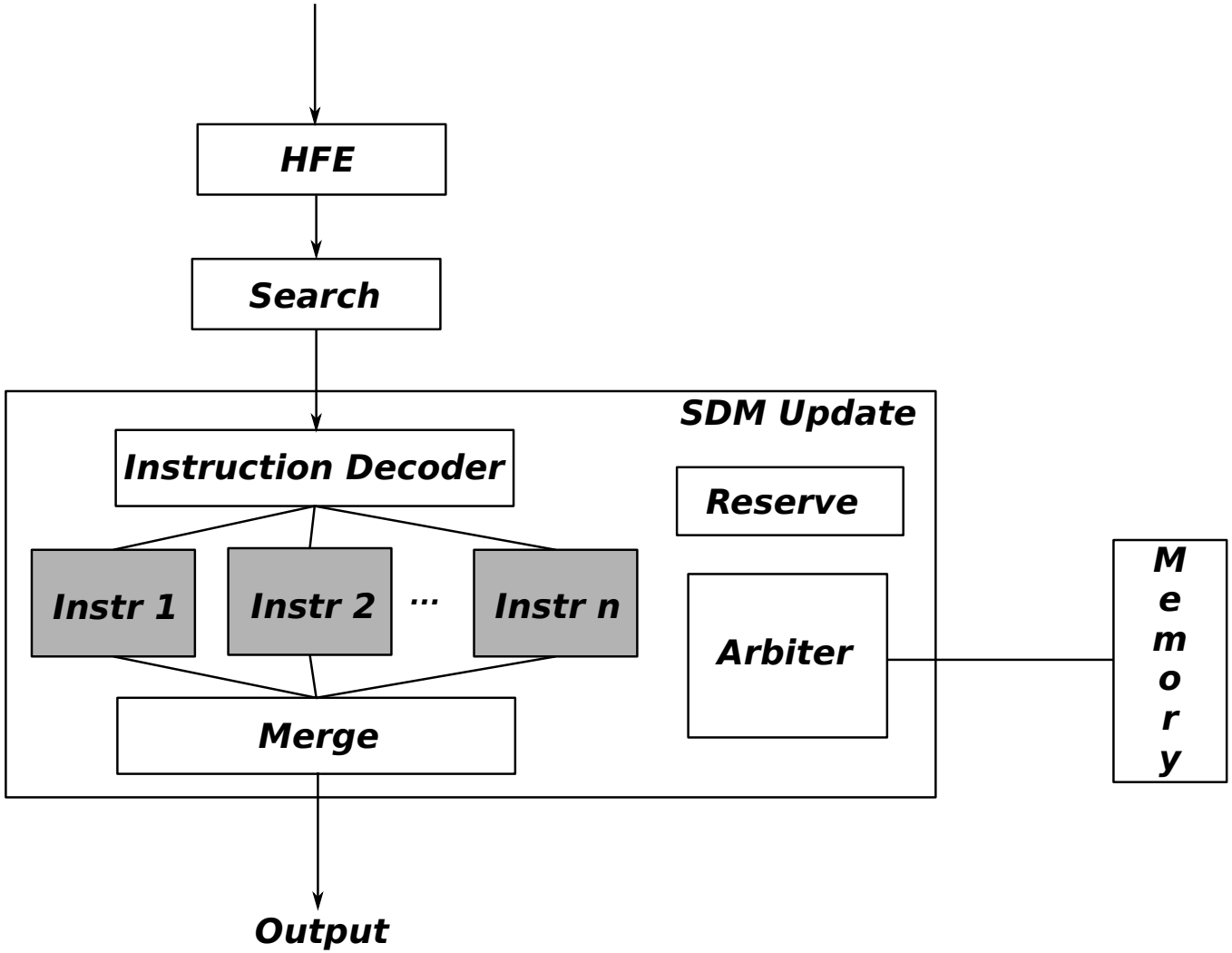
- needs payload of selected frames, but do not have to see all incoming frames
  - **default:** use *Send* on interesting traffic, *Drop* the rest
  - **rules:** *Drop* rules for already processed flows
- CPU performance savings:
  - processing of interesting flows only
  - not all packets from interesting flows must be processed
- easy deployment in combination with UC1

UC2: Results



- **HTTP**: $\frac{1}{4}$ of frames and $\frac{1}{4}$ of data load
- **DNS**: $\frac{1}{100}$ of frames and $\frac{1}{200}$ of data load

UC3: SDM Firmware as Processor



UC3: Monitoring Instructions

- update of stored record based on the frame data
  - consist of operation code and record address
  - delimited by 2 memory accesses (read and write back)
  - update process can vary
- new instructions without changes in existing modules
- new instructions created in C/C++ with HLS
  - consumes less time and allows faster implementation
  - verification during implementation
  - even software guy can create accelerated solutions

UC3: Demonstration Instructions

- **NetFlow (I1)**
  - basic NetFlow aggregation (basic Aggregate)
  - packet/byte counters, start/end timestamps, TCP flags
  - part of the basic SDM infrastructure
- **NetFlow Extended (I2)**
  - I1 with TCP flags of the first 5 packets of the flow
  - demonstrates easy NetFlow extending using plain C
- **TCP Flag Counters (I3)** (Non-NetFlow)
  - counts the number of observed TCP flags
  - support advanced flow analysis
- **Timestamp Diff (I4)** (Non-NetFlow)
  - inter-arrival times of the first 11 packets
  - flow based classification or identification of L7 protocols

UC3: Results

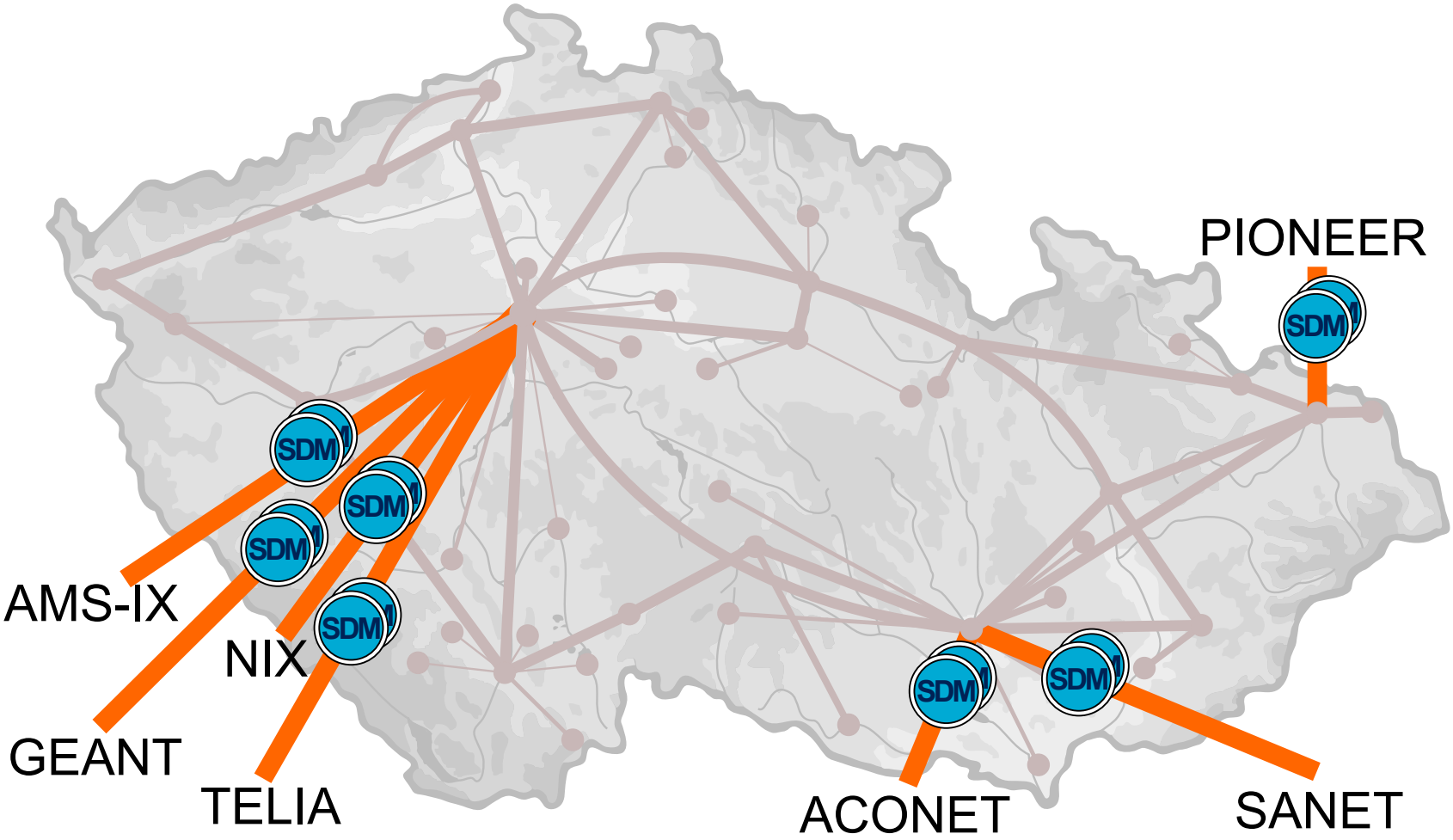
Instruction	Regs	LUTs	Freq. [MHz]
(I1)NetFlow (handmade)	1754	325	425.134
(I1)NetFlow	1846	824	308.641
(I2)NetFlow Extended	2070	1113	308.641
(I3)TCP Flag Counters	0	1046	327.868
(I4)Timestamp Diff	5199	2556	306.748

- all modules meet the frequency requirement for 100 Gb/s
- HLS do not beat hand-written VHDL, but is good enough
- instruction creation in C/C++ is very simple and fast
- even non-VHDL programmer can accelerate his monitoring

New Metering Point

- commodity server running Linux
- SW flow exporter (NetFlow/IPFIX) from SME Invea-Tech
  - support for creation of traffic processing plugins
  - plugins utilizing the SDM acceleration capabilities
- our own hardware probe for up to 100 GbE
  - new PCI-Express card with powerful Virtex7 FPGA
  - 1×100 GbE or 2×40 GbE or 8×10 GbE interfaces
  - SDM over NetCope as firmware

Future NREN Cesnet



- all metering points doubled (production and testing)

Thank you for your attention.