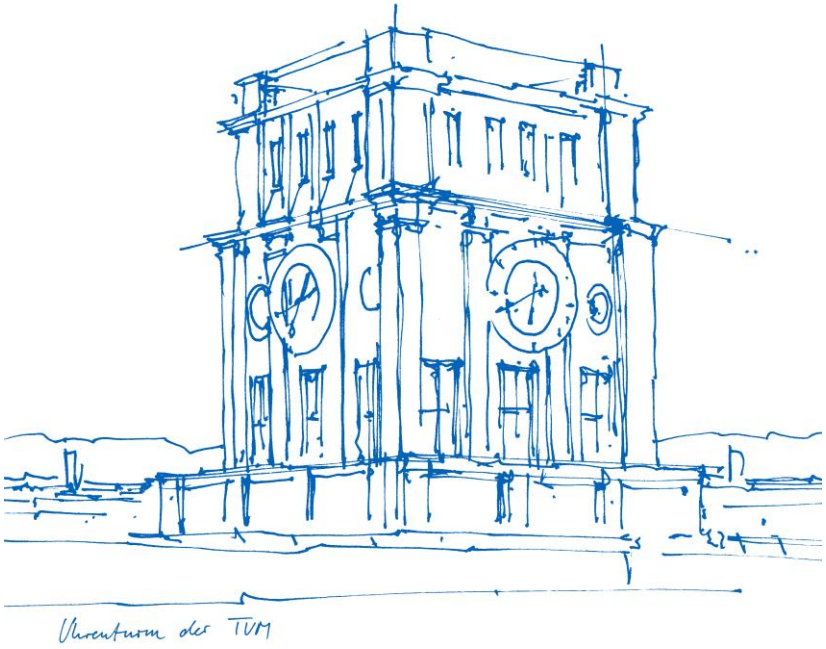
Self-Aware Load Distribution for Software-Defined Network Nodes

Andreas Oeldemann, Thomas Wild, Andreas Herkersdorf

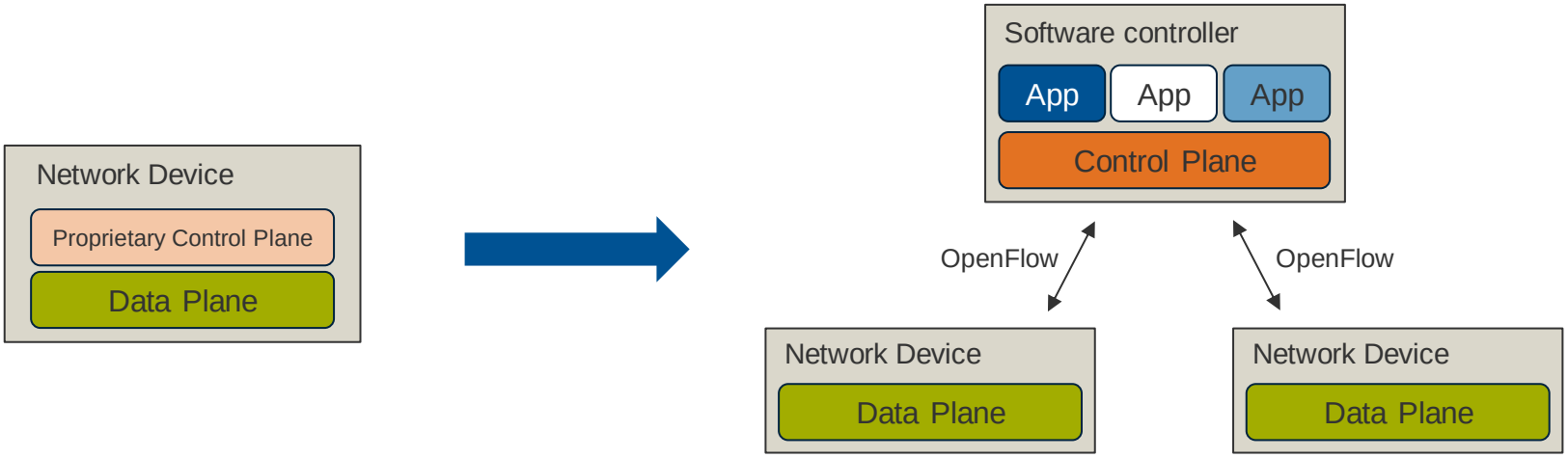
Technical University of Munich



MPSoC Forum 2016, Nara, Japan
July 14th, 2016

Changing Network Landscape

Software-Defined Networking



- Network knowledge and control is moved to (logically) centralized controller
- Open control interfaces (e.g. *OpenFlow*)
- Programmable and automatable control

Changing Network Landscape

Network Function Virtualization

- Replacement of specialized network appliances by standard x86 servers
- Any device can execute any arbitrary compute or network application

Performance?

Energy efficiency?

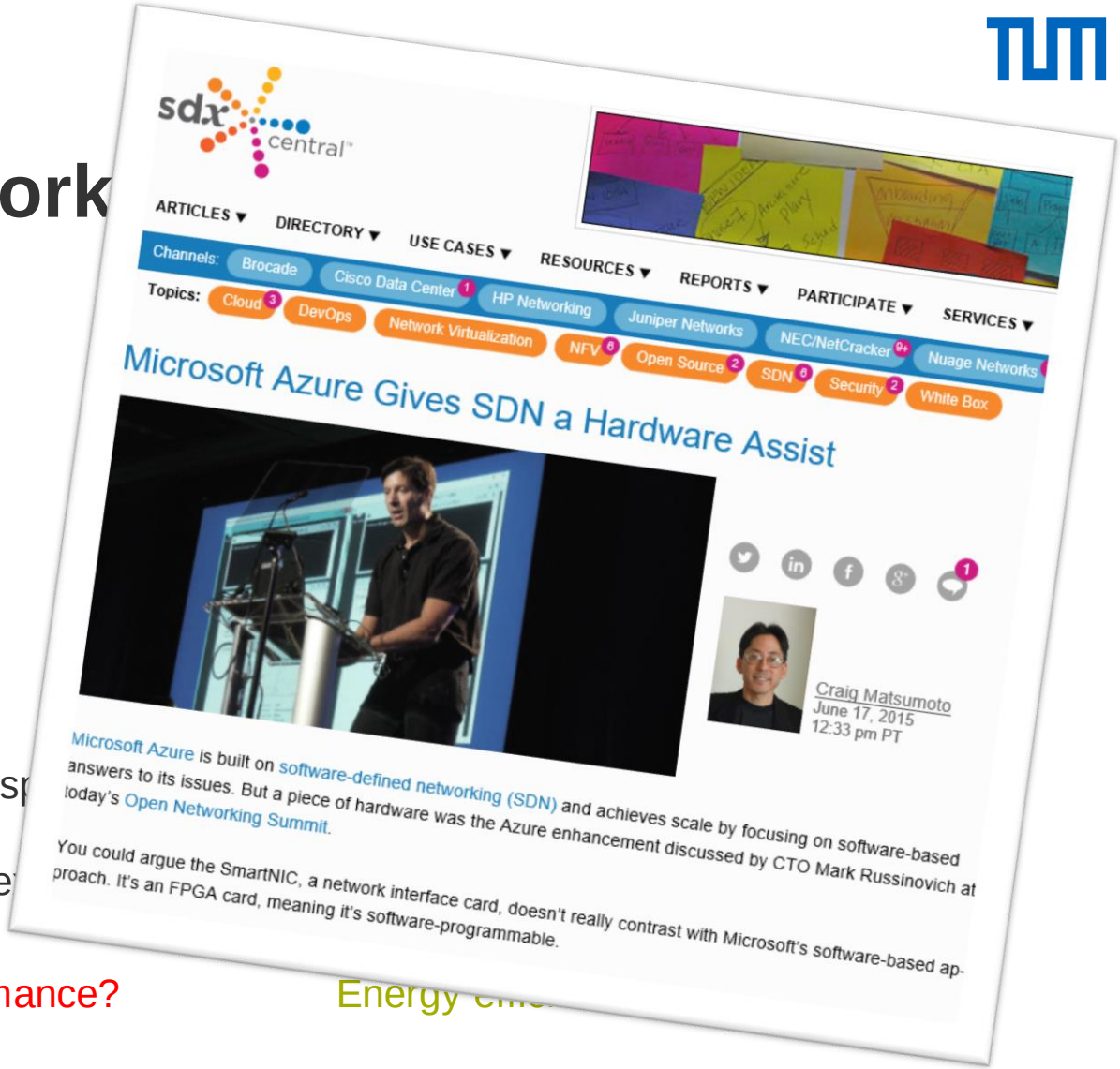
Changing Network

Network Function Virtualization

- Replacement of sp
- Any device can e

Performance?

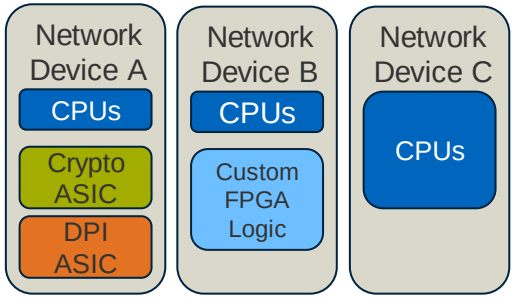
Energy effi



We believe that network processing will be software-centric, but heterogeneity in processing platforms will prevail…

… and we are not alone!

Network Heterogeneity



Three types of heterogeneity:

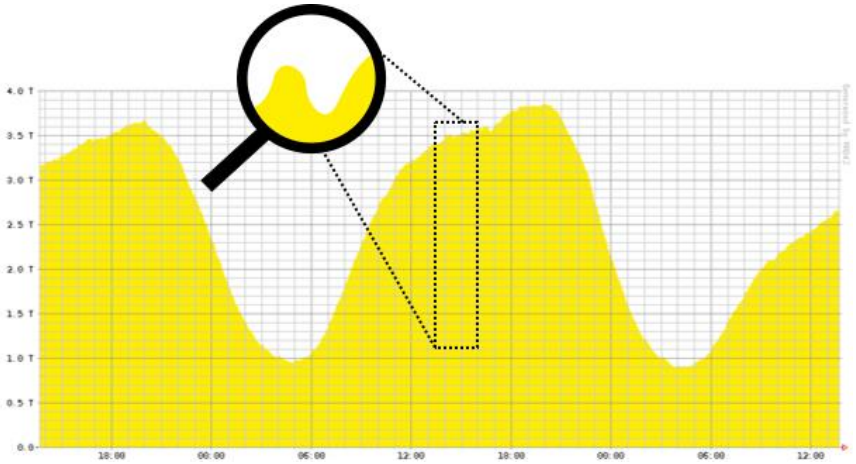
1. Network devices with heterogeneous processing architectures (e.g. NPUs)
2. Heterogeneous composition of network devices across network
3. Network applications with different processing complexities and hardware acceleration opportunities (e.g. NAT, DPI, compression, …)

How can centralized network control exploit network resources efficiently?

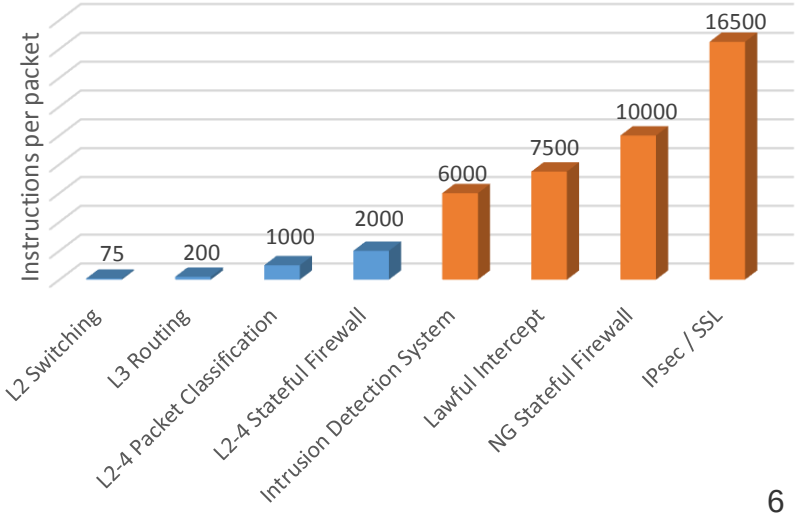
Centralized Control Problems

Problem 1: Traffic Dynamicity

- Internet traffic data rate and mix varies significantly during run-time
  - Time of day, large sport events, …
- Processing demands of network equipment depends on both data rates and network function processing complexity
- Fluctuations in traffic composition (e.g., bursts) may lead to rapid increase of temporary necessary processing resources

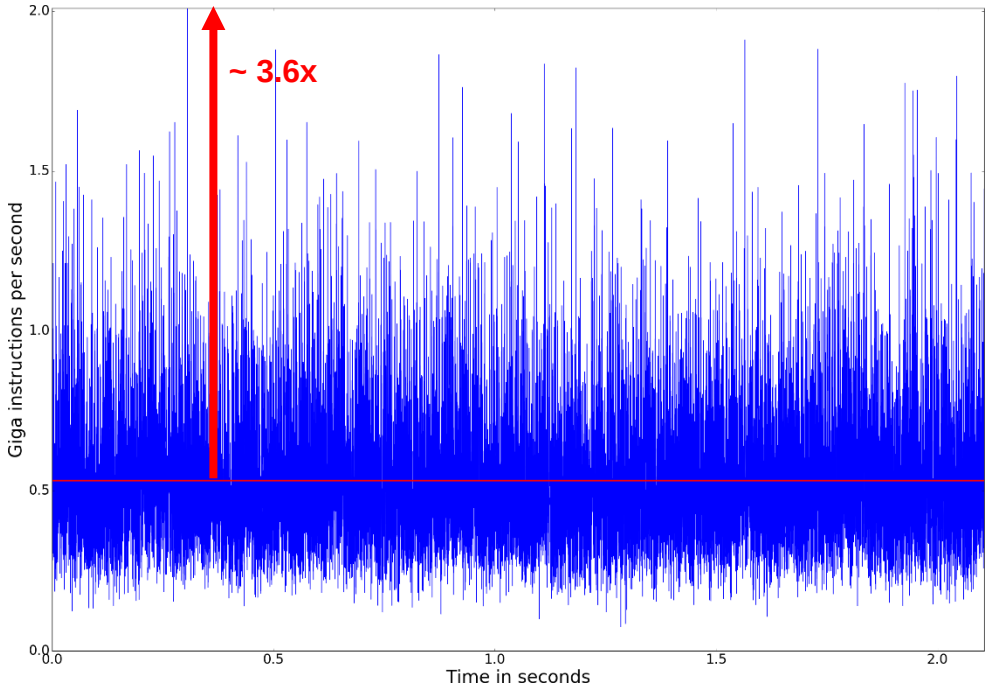


DE-CIX Frankfurt average traffic in bits per second on March 14-16 2015, Source: DE-CIX



Centralized Control Problems

Problem 1: Traffic Dynamicity



Sample Time: 100 us

Network Trace Input:

- 10 Gbps backbone link (~25% loaded)

Network Functions Assumptions:

	IPP	Traffic share
VNF 1	100	40%
VNF 2	5000	10%
VNF 3	900	25%
VNF 4	1500	25%

- Network functions randomly assigned to /16 IPv4 and /64 IPv6 destination prefixes

Network control latencies limit real-time NE function scaling during short-term bursts

Source: The CAIDA UCSD Anonymized Internet Traces 2015 – 2015-12-17, http://www.caida.org/data/passive/passive_2015_dataset.xml

Short-Term Traffic Bursts

How to handle these short-term fluctuations in processing demands?

Option A: Overprovisioning

- Provision more resources than required to process the average load

Resources idle for large share of time, cost prohibitive

Option B: Buffering

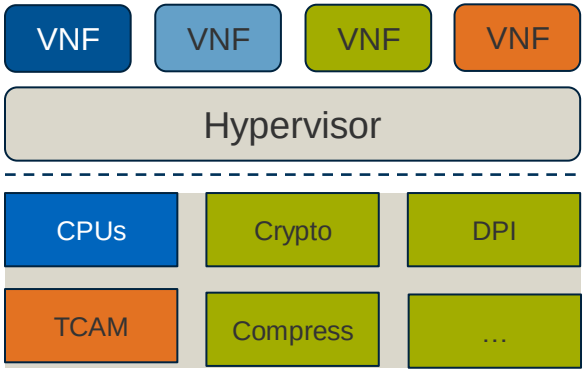
- Buffer packets until sufficient resources are available to process them

Higher latency, latency jitter

Centralized Control Problems

Problem 2: Data Plane Visibility

Efficient utilization of processing resources requires extensive knowledge about platform implementation:

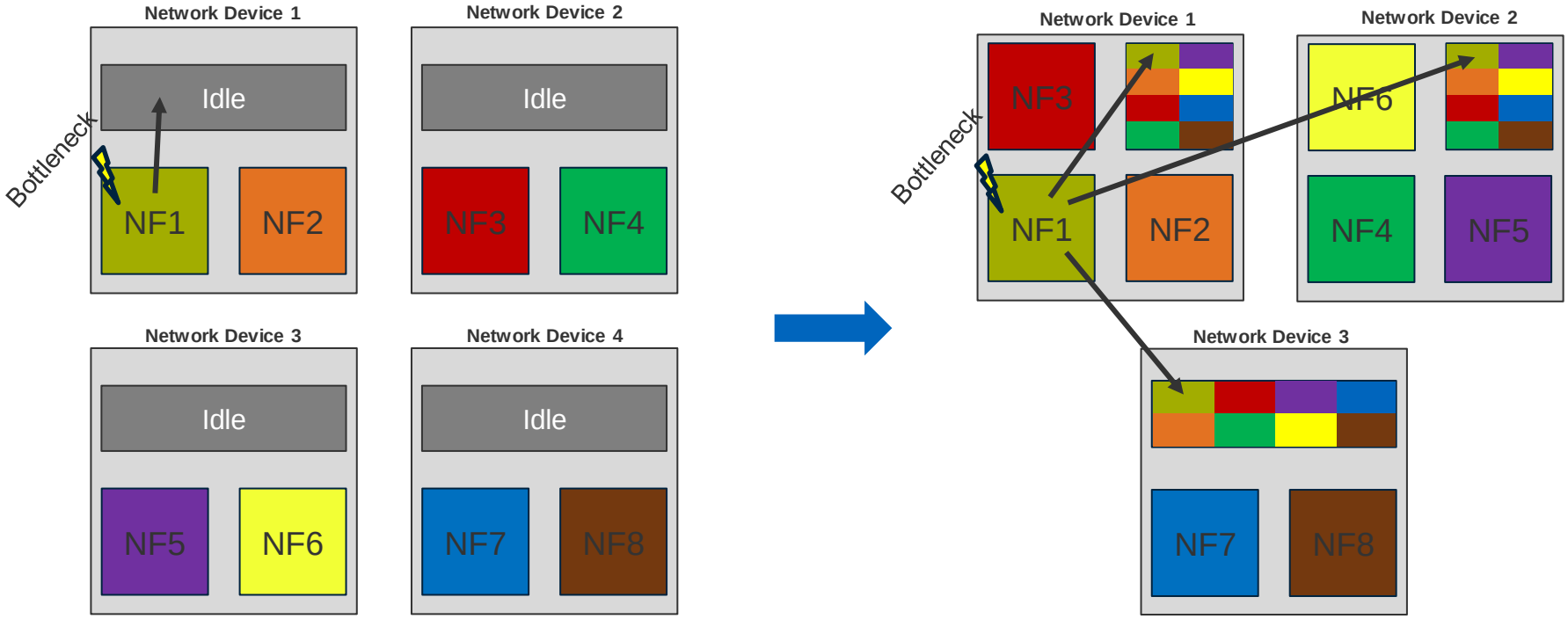


- Which accelerators are available in each NE?
- What are key performance/energy characteristics of each NE?
- How are VNFs mapped to hardware resources?
- How are resources shared among VNFs?
- In case of a resource bottleneck, which VNFs or traffic flows are responsible?

Huge amounts of rapidly changing, platform-specific knowledge required for controller decision making under global awareness

Accurate control only possible when vendors are willing to reveal corresponding product details / intellectual property.

Overprovisioning Alternatives



Traditional:

- Processing peaks are handled by utilizing locally over-provisioned resources

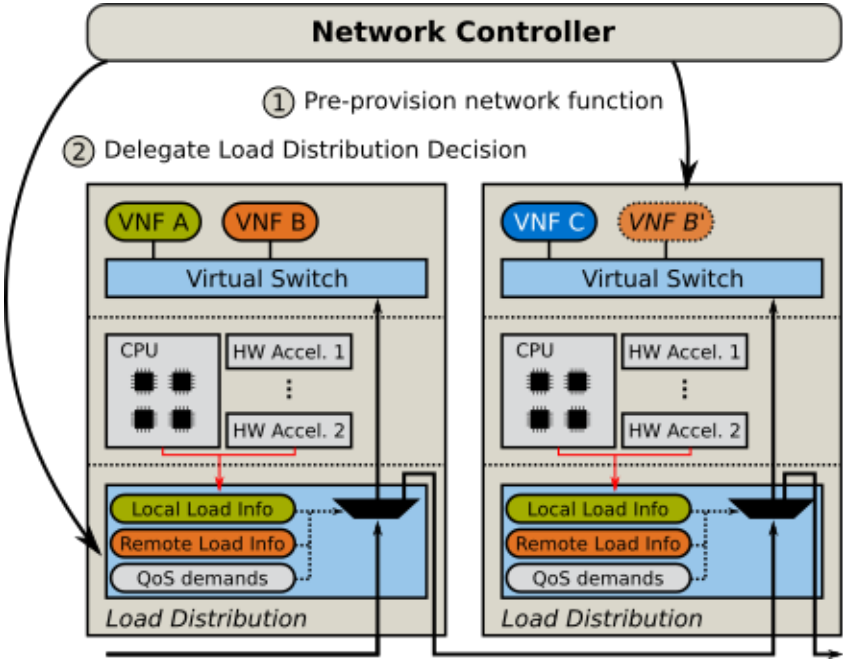
Our Approach:

- Utilize spare resources on other devices by self-aware redirection of peak loads
- Lower overall resource demand due to sharing effects (claim to be proven)

Self-Aware Load Distribution

1. **Network function pre-provisioning**
   - Proactive provisioning of network function duplicates in NEs to prevent large instantiation delays

2. **Control Delegation for Load Distribution**
   - NEs may perform packet flow distribution on behalf of, but without reactive reassurance with, the Network Controller

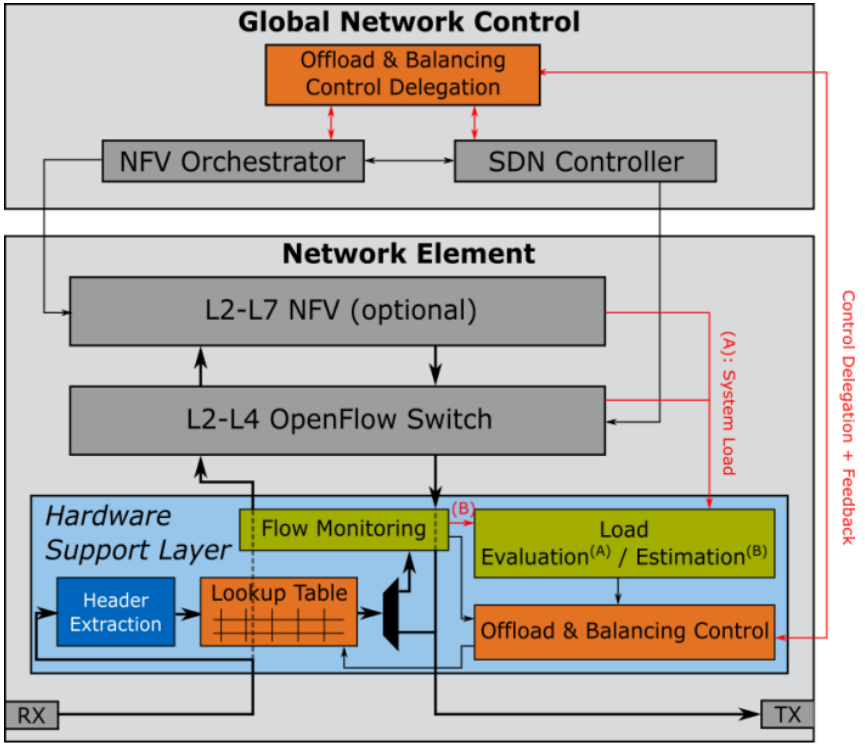


3. **Self-Aware Wirespeed NE Load Distribution**
   - Hardware-accelerated per packet decision making whether processing shall be performed at local NE or redirected to an alternative NE …
   - … in awareness of local and remote resource availability

Self-Aware Load Distribution

Hardware Support Layer (HSL)

- **local** system load (measured and estimated)
- load of **neighbor** network nodes (state information exchange)

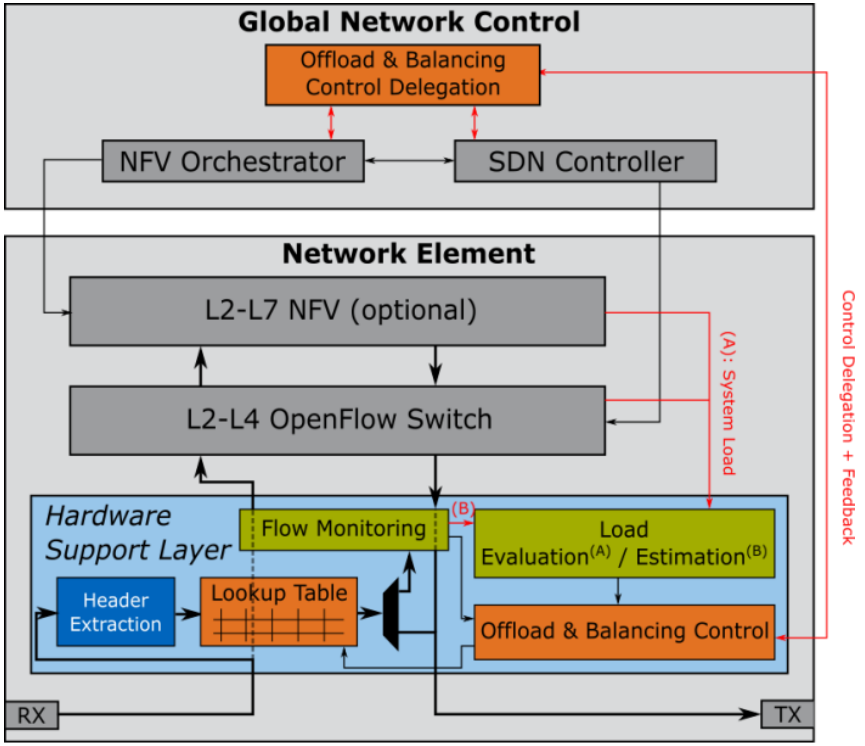


- HSL optimizes local resource utilization by distributing excess traffic to neighbor NEs
- NEs jointly work towards overall optimization of resource utilization and fulfilment of QoS demands specified by the controller
- HSL closely-coupled with NE data plane hardware → no control latencies to remote network controller

Self-Aware Load Distribution

Hardware Support Layer (HSL)

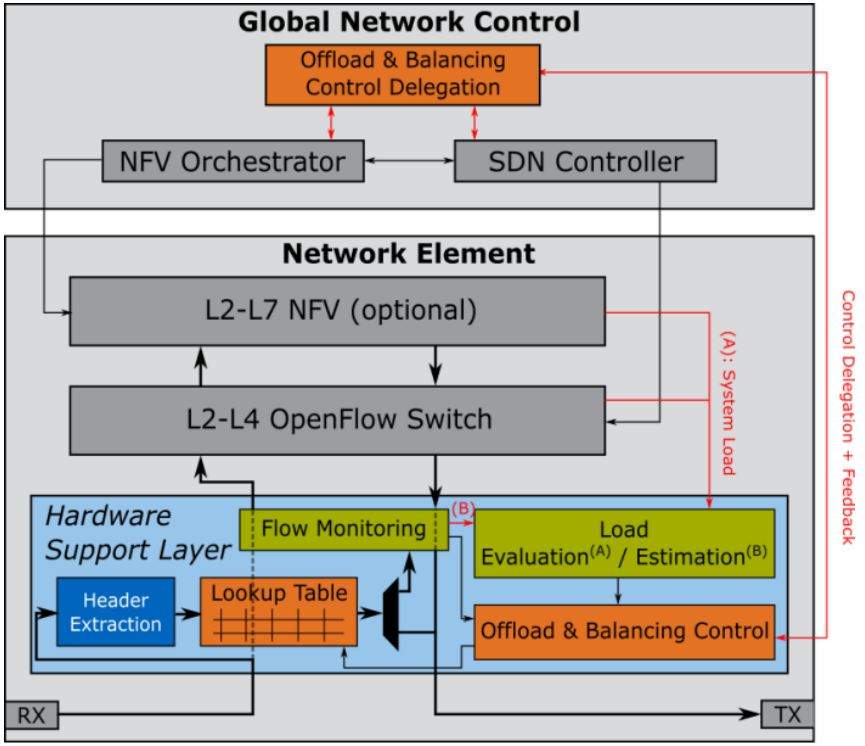
- **Header Extraction / Lookup:** Wirespeed packet parsing, destination extraction and LUT-based classification
  - Deep pipelined, wide data word packet handling for high throughput (512bit @ 100 MHz for 40Gbps link)
  - Classification result decides about local / remote packet processing



Self-Aware Load Distribution

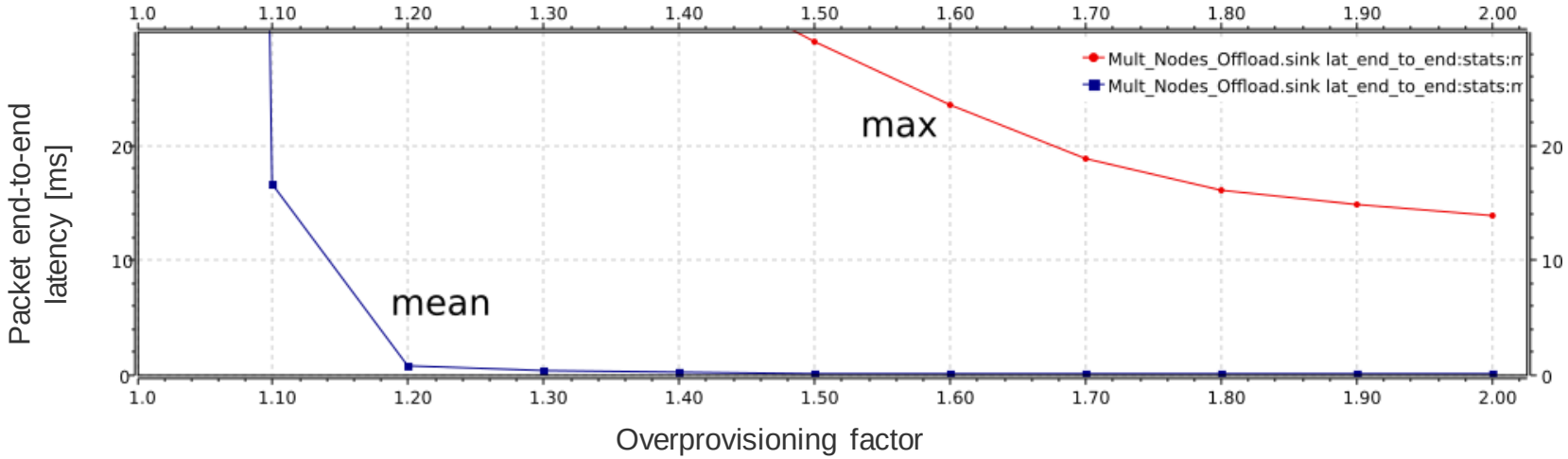
Hardware Support Layer

- **Header Extraction / Lookup:** Wirespeed packet parsing, destination extraction and LUT-based classification
- **Flow Monitoring:** Gathering statistics on packet rates and latencies for locally processed flows / flow bundles
- **Load Evaluation / Estimation:** Self-aware projection of NE utilization based on Flow Monitoring and NE performance counters
- **Offload & Balancing Control:** LUT updates to reflect local load situation

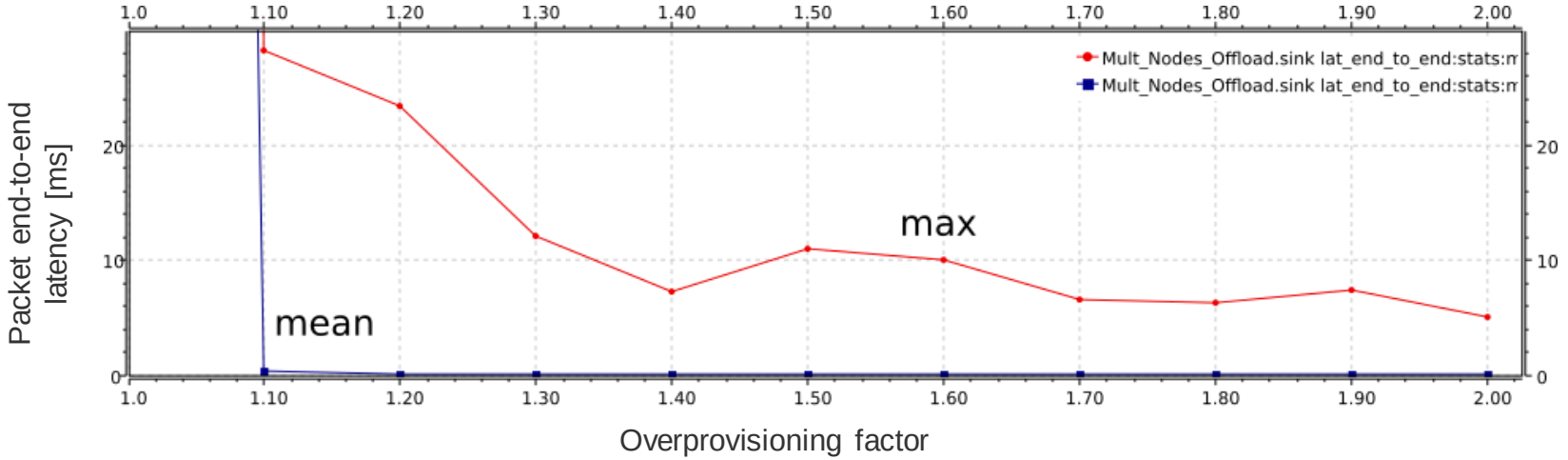


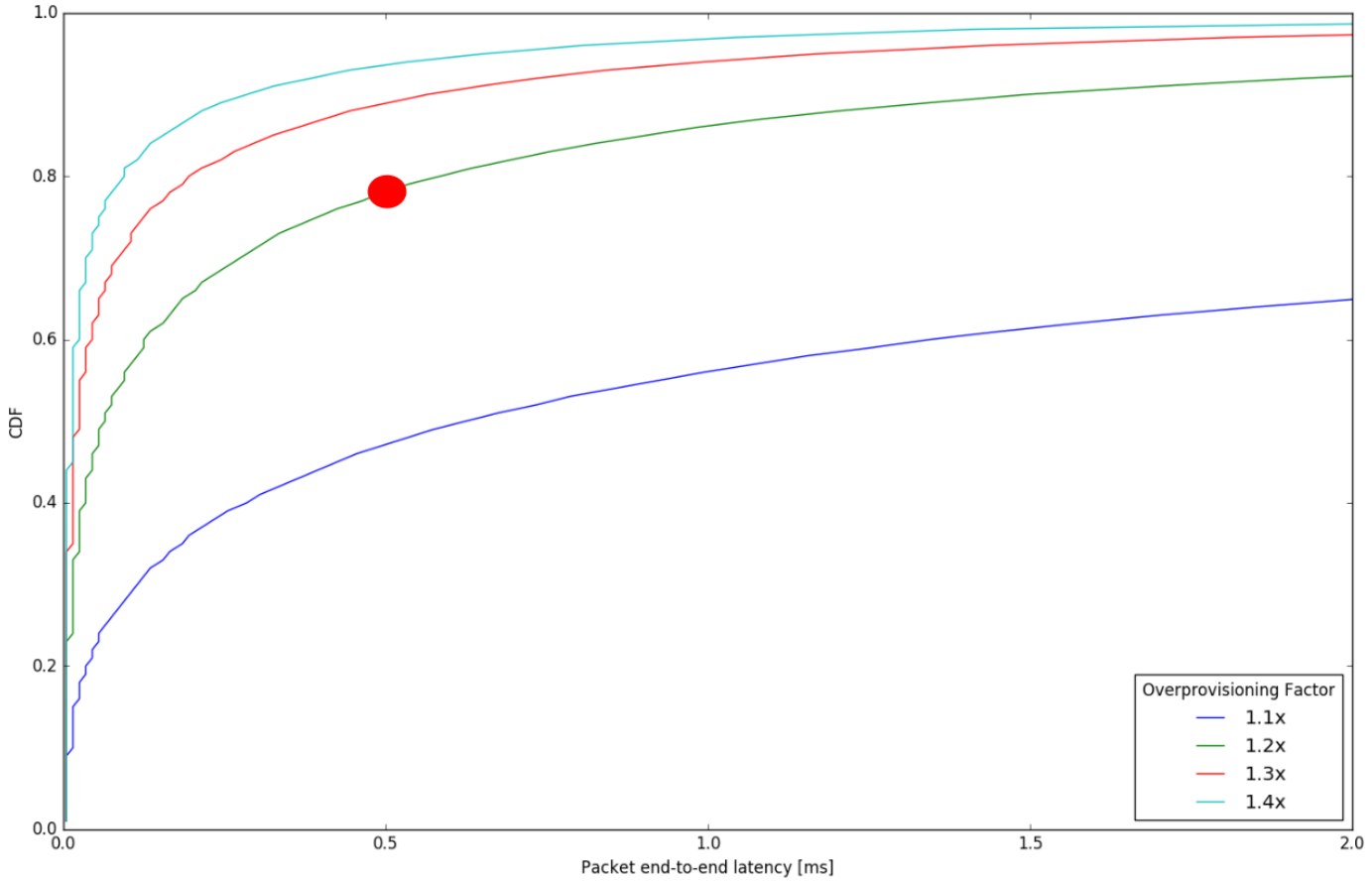
Higher responsiveness due to reduced control latency; Lower over-provisioning
Improved accuracy in decision making

Load distribution disabled



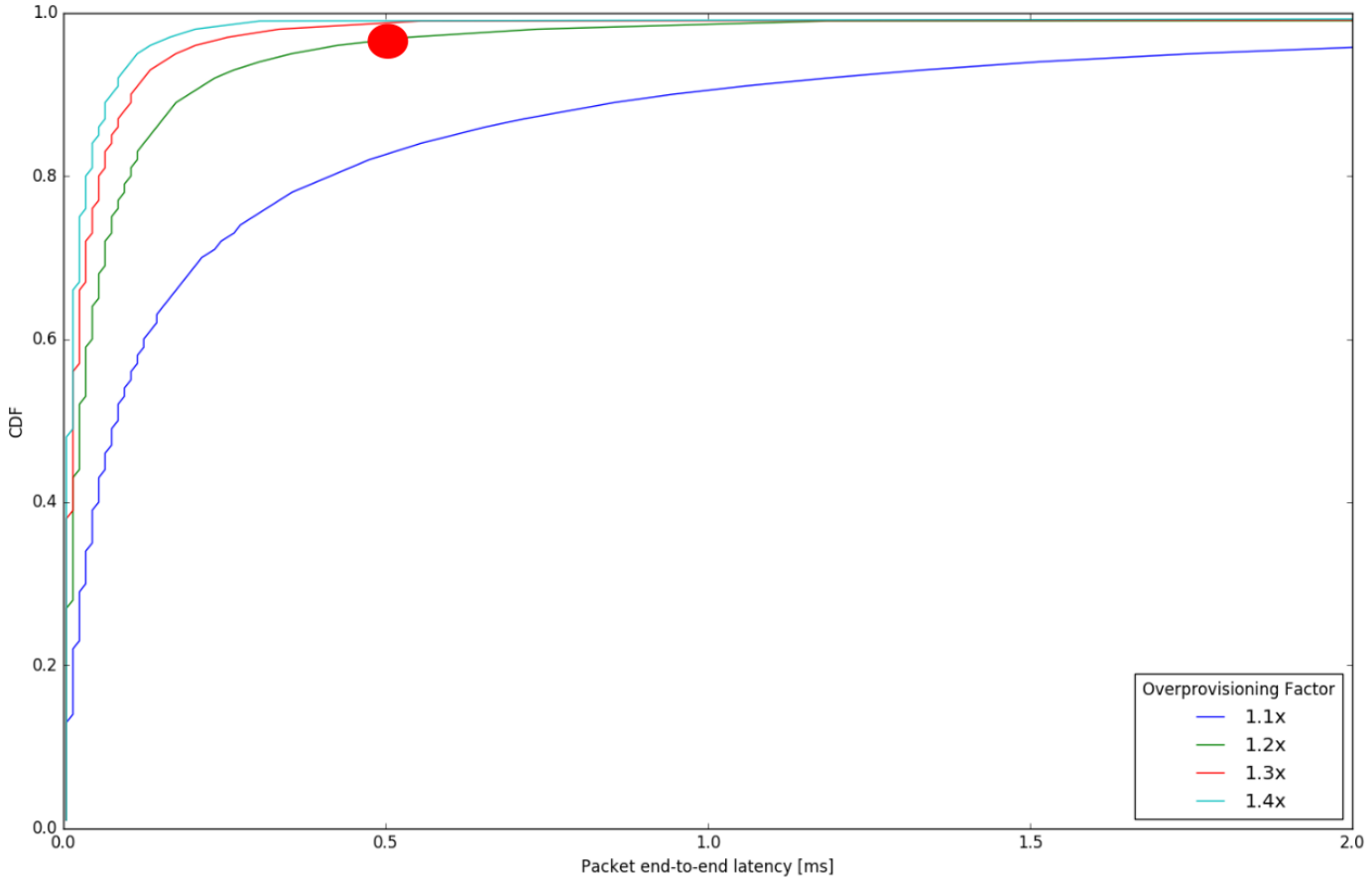
Load distribution enabled





Load distribution **disabled:**

With 20% overprovisioning, ~75% of packets with latency below 0.5 ms



Load distribution **enabled:**

With 20% overprovisioning, ~95% of packets with latency below 0.5 ms

Conclusion

- Spatial separation of network control and data planes in SDN/NFV comes with problems:
  - Control latencies make reaction to traffic bursts and, in general, short-term increases in processing demands infeasible
  - Network heterogeneity (devices, platforms, applications) prevents efficient exploitation of data plane processing resources without huge amounts of device- and application-specific knowledge in controller
- Our work-in-progress approach proposes a self-aware hardware support layer to mitigate such problems by …
    - … allowing the network nodes themselves to take proactive actions for resolving resource shortages and …
    - … dynamically distribute processing tasks based on situation-dependent availability of resources in local and neighboring network nodes

Thanks / Questions?

Contact

Andreas Herkersdorf <herkersdorf@tum.de>
Institute for Integrated Systems, Technical University of Munich, Germany