Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

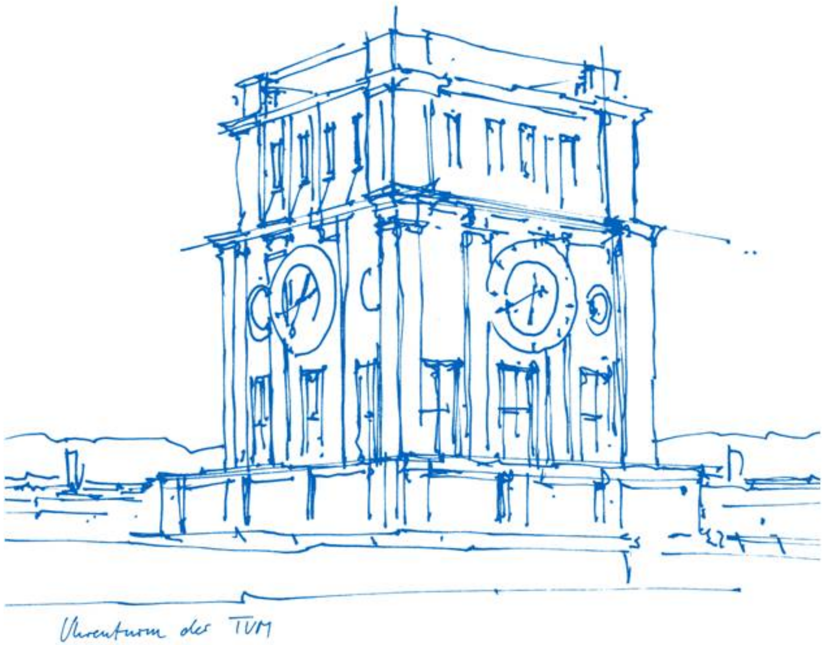
GEFI 2019
Reproducibility by Design: A Family of Testbeds for High-Precision Network Experiments

Georg Carle
carle@net.in.tum.de
http://www.net.in.tum.de/~carle

Sebastian Gallenmüller
gallenmu@net.in.tum.de
http://www.net.in.tum.de/~gallenmu/

8 November 2019

Acknowledgements:
Members of the Chair of
Network Architectures and Services



Garching b. Munich, Research Campus

Outline

Challenges

Approach

- Testbed for reproducible experiments
- Flexible workload generation
- Selected results

Next steps

Conclusions

Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

Challenge:
Reproducible Network Experiments

Reproducible Experiments

ACM SIGCOMM MoMeTools - Workshop on Models, Methods and Tools for Reproducible Network Research
Georg Carle, Hartmut Ritter, Klaus Wehrle,
Karlsruhe, Germany, August 2003

ACM SIGCOMM Reproducibility Workshop
Olivier Bonaventure, Luigi Iannone, Damien Saucez
Los Angeles, USA, August 2017

[Rep17] Q. Scheitle, M. Wählisch, O. Gasser, T. Schmidt, G. Carle,
Towards an ecosystem for reproducible research in computer networking
Proceedings of the ACM SIGCOMM Reproducibility Workshop, 2017

Dagstuhl seminar 18412 “Encouraging Reproducibility in Scientific Research of the Internet“, October 2018

Despite 16 years since first workshop have passed, issues remain

- What influences the performance of networked systems?
- Which KPIs are relevant?
- How to measure these KPIs?
- How to build experiment setups measuring these KPIs?
- How to measure in a ***reproducible*** manner?

Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

Challenge:
High-Quality Data

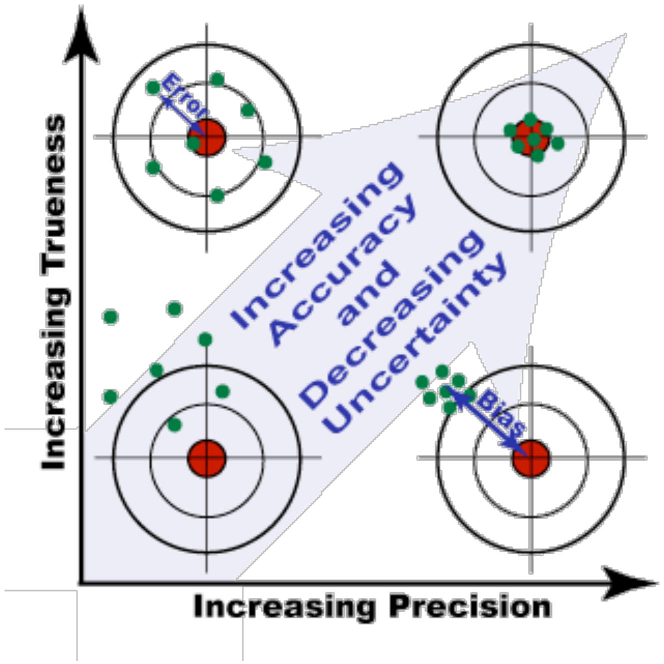
Dimensions of Data Quality: Precision and Accuracy

Precision

- **Random errors** in the generation process
- Traffic generator: How much do individual inter-packet gaps deviate from the configured value?

Accuracy/Trueness

- **Systematic errors (bias)** of the generation process
- How close is the average observed rate to the configured one?



ISO 5725-1, “Accuracy (trueness and precision) of measurement methods and results – Part 1: General principles and definitions”.

Figure see:

Nondestructive Evaluation (NDE)
https://www.nde-ed.org

Dimensions of Data Quality: Coverage of System States

Coverage in collected data

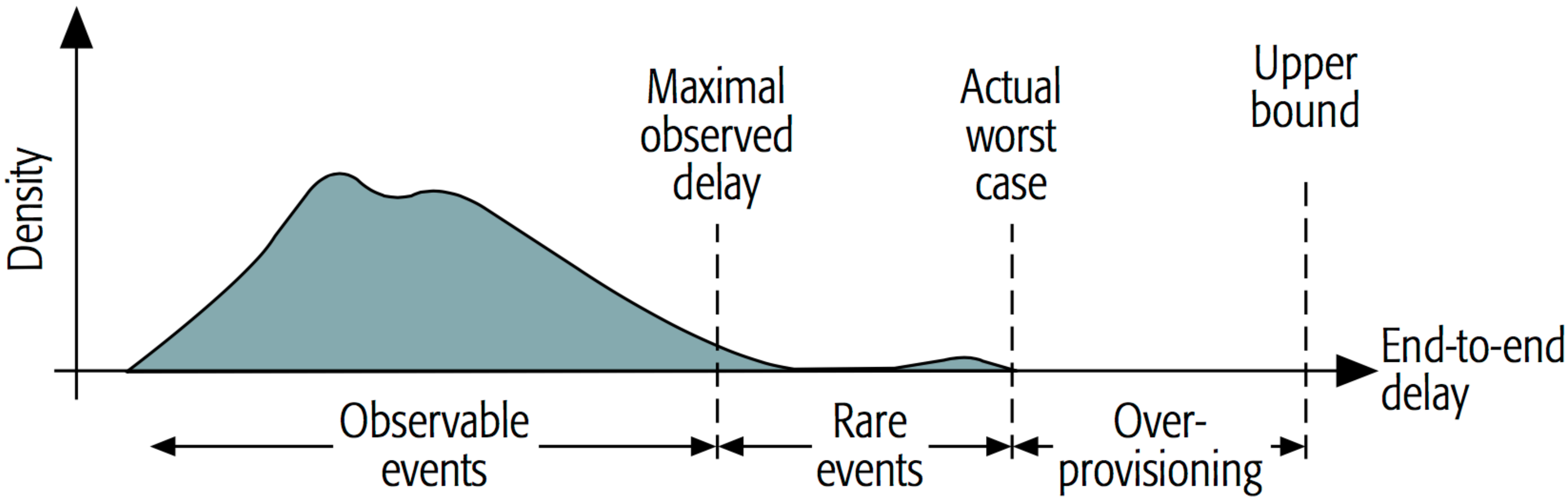
- Average cases
- Rare cases

Challenge

- Are relevant system states sufficiently covered in the observed data?

Example for Coverage: End-to-End Delay

Maximal observed delay vs. upper bound



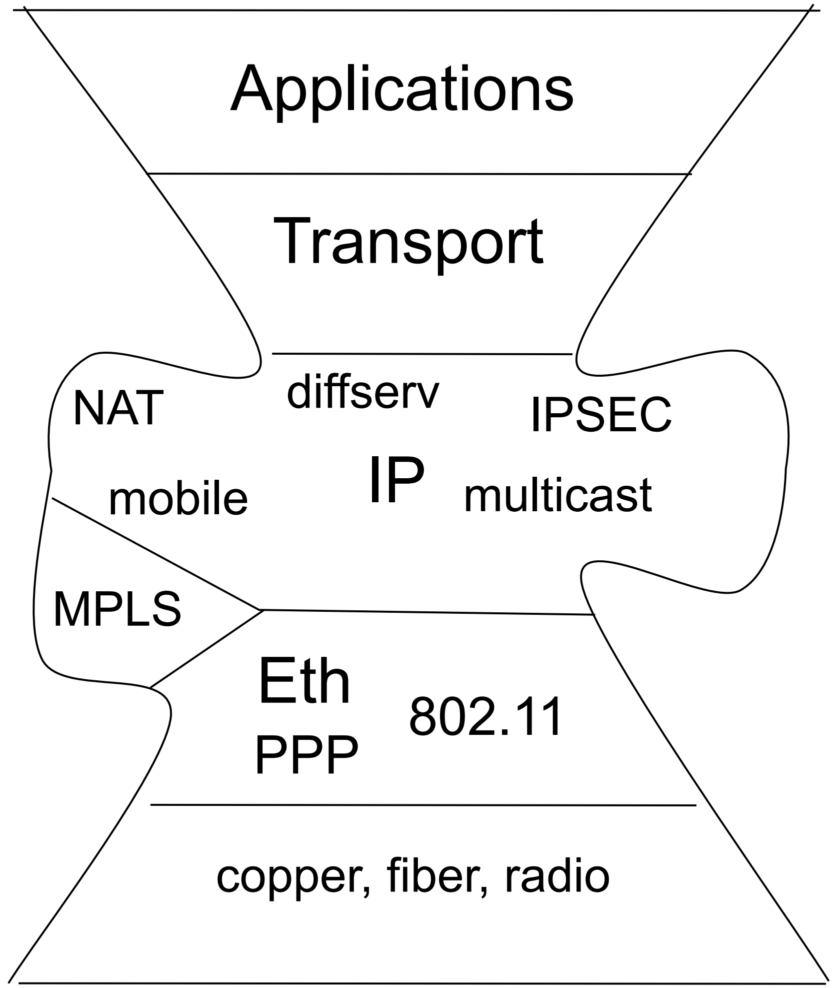
[ComMag16] Fabien Geyer, Georg Carle: Network engineering for real-time networks: Comparison of automotive and aeronautic industries approaches, IEEE Communications Magazine 54 (2), 2016

Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

Challenge: Complexity

Protocol Stacks are Complex

- TLS, DTLS
- TCP, UDP, SCTP, DCCP, QUIC
- BGP, OSPF, IS-IS, RIP, RIPng, VRRP, PIM, IGMP, MLD
- IPsec, IKE, EAP
- IPv4, IPv6, ICMP
- VLAN, GTP, IP in IP, GRE, L2TP, MPLS



Protocol Implementation Trends add Complexity

Hardware trends

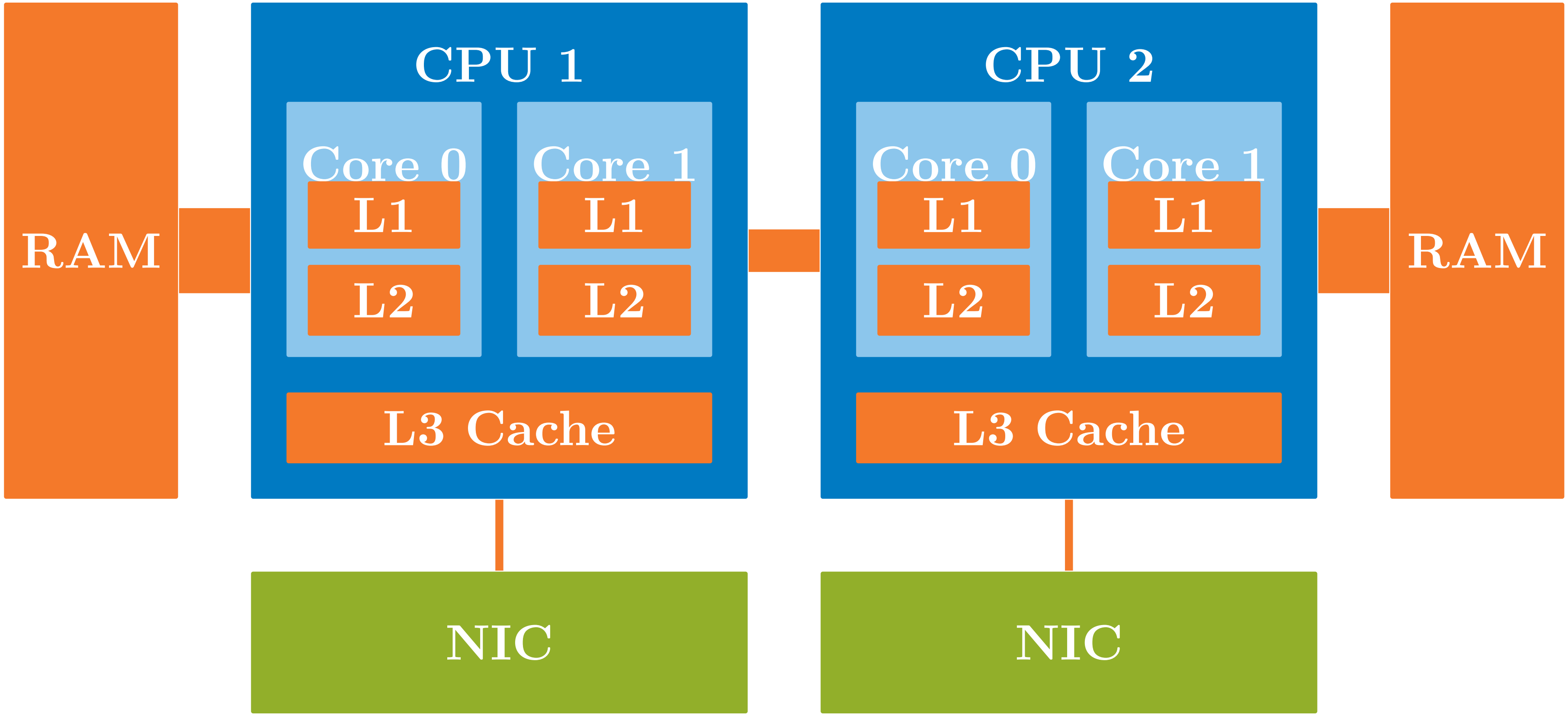
- Multi-core/many-core CPUs
- Multi-queue NICs
- Programmable NICs
  - Netronome SmartNIC with Network Flow Processor (NFP)
- Programmable Switches
  - Tofino P4 Switch

Software trends

- High-performance packet processing frameworks
  - DPDK, netmap, Snabb
- Virtualization
  - Xen, KVM
- Containers
  - namespaces, cgroups

Modern Hardware Architectures are Complex

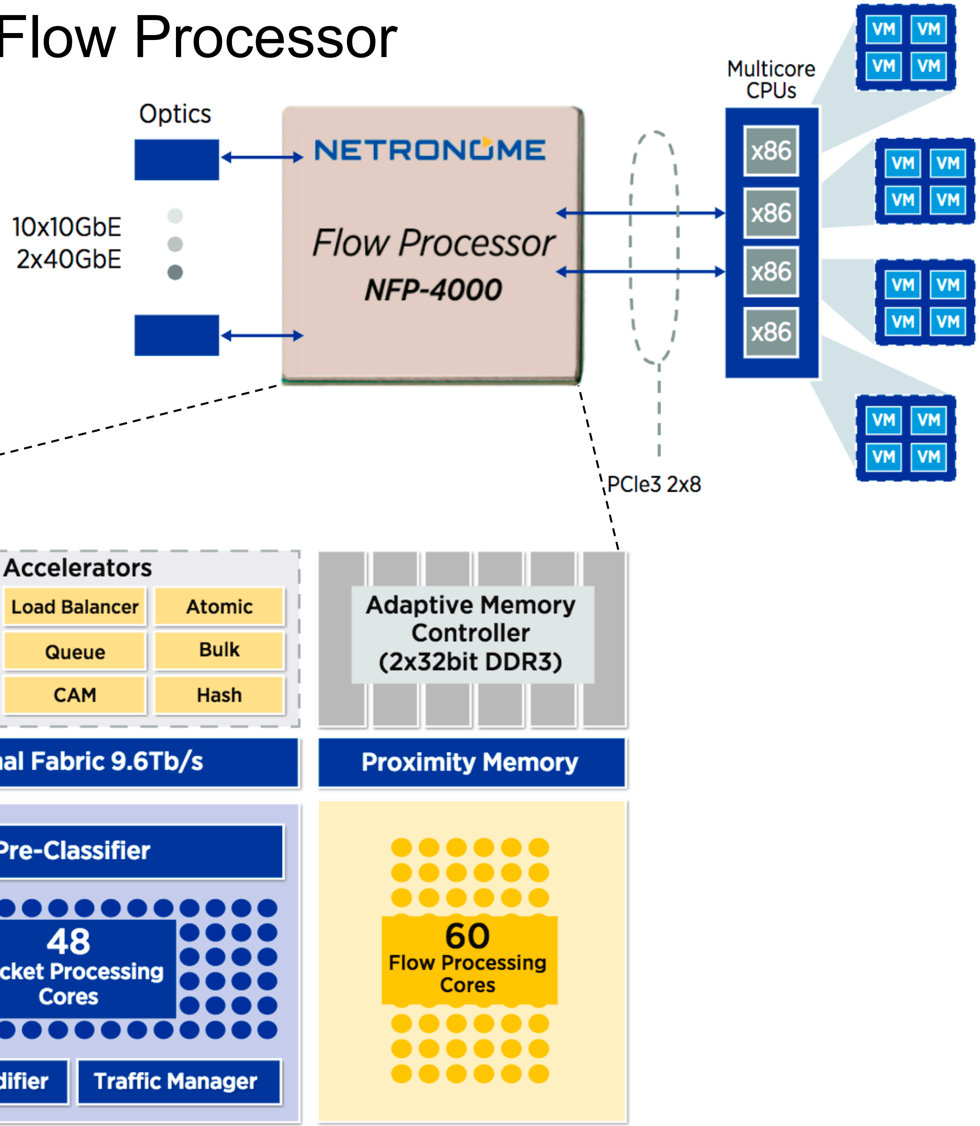
Non-Uniform Memory Architecture (NUMA)



Programmable NICs add Complexity

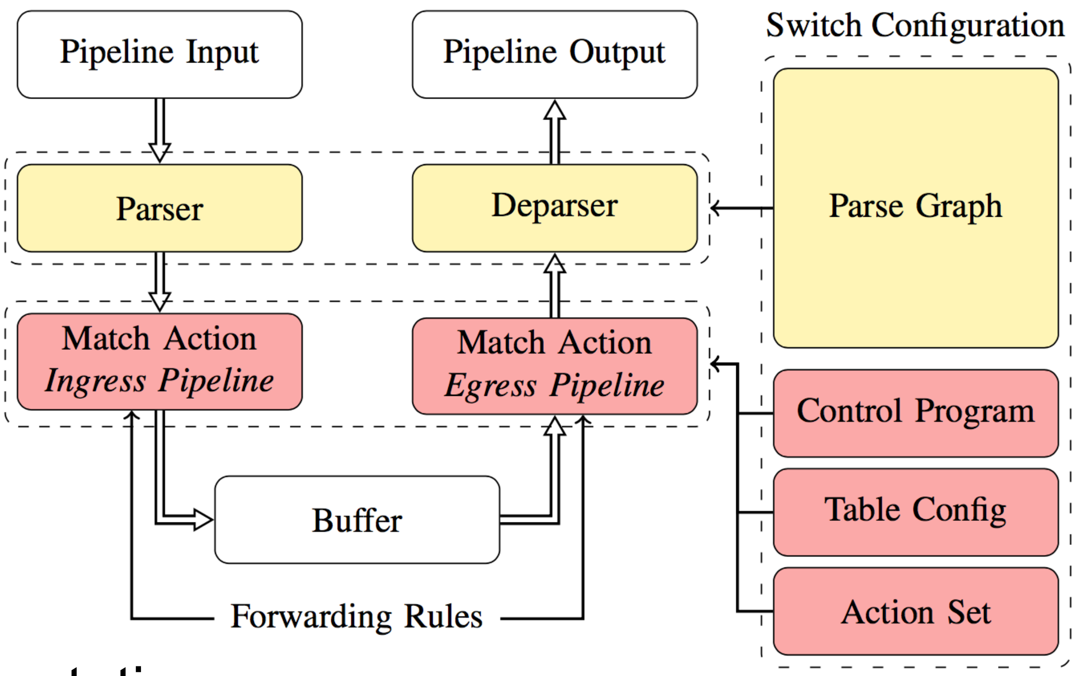
SmartNIC

- Netronome with NFP-4000 Flow Processor
  see www.netronome.com



Programmability adds Complexity

P4 Architecture



P4 Hardware Implementation

- Tofino switch
- P4NetFPGA

P4 Software Implementations

- P4@ELTE based on DPDK
- PISCES based on Open vSwitch with DPDK

Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

Challenge

Mastering Software Complexity

High-Level Languages for Improved Low-Level Software

Security Bugs in Operating Systems

- 1999: Linux 2.2.0: 1.2 M lines of code; driver code: 54%
- 2009: Linux 2.6.29: 6.9 M lines of code, driver code: 53%
- 2019: Linux 4.19: 17 M lines of code; driver code: 66%
- 97% of security bugs related to memory safety found in Linux in 2017 are located in drivers.

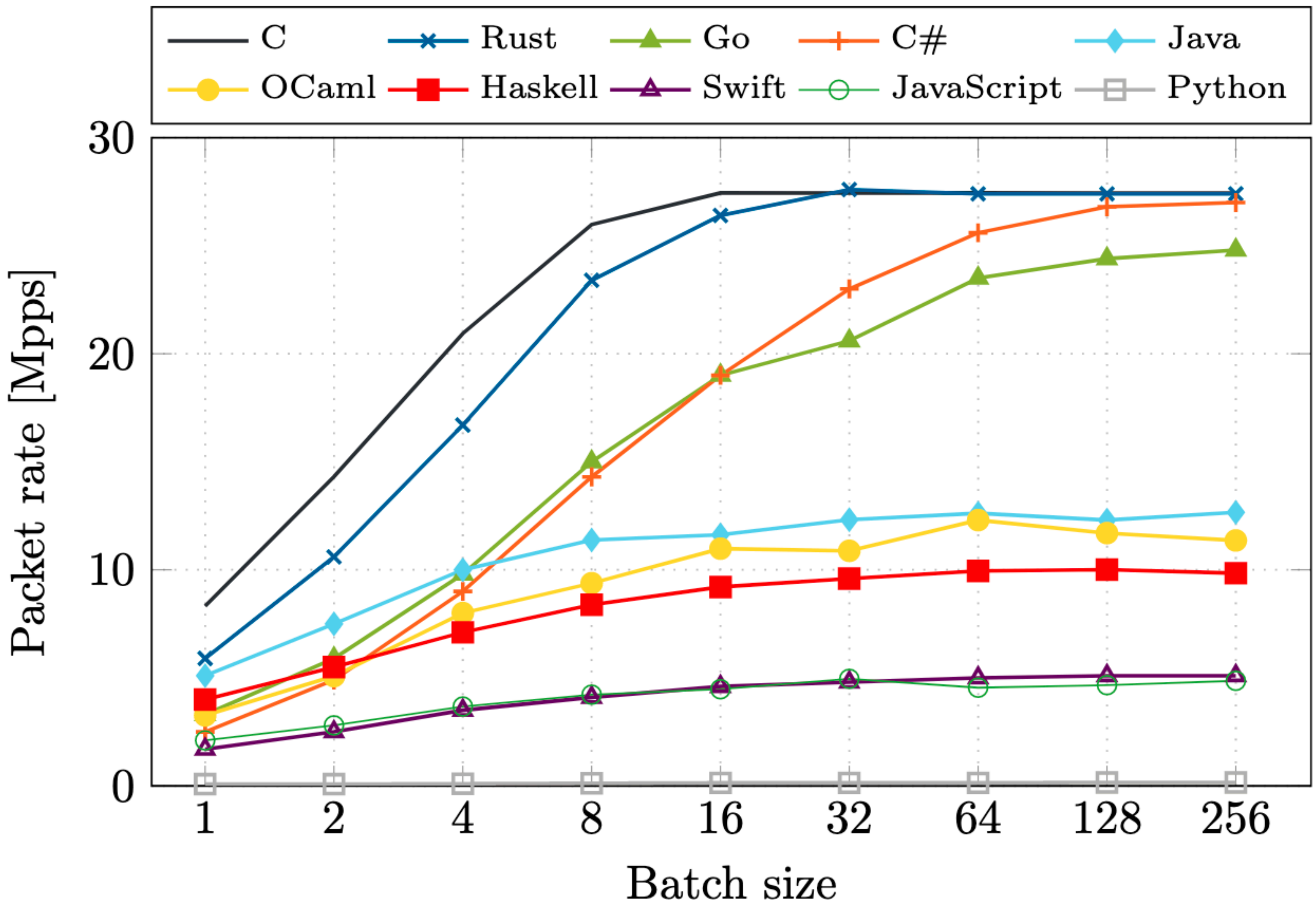
Potential for improvement

- Using high-level programming languages for drivers
- User-space drivers

[ANCS2019a] P. Emmerich, S. Ellmann, F. Bonk, A. Egger, E. Sánchez-Torija, T. Günzel, S. Di Luzio, A. Obada, M. Stadlmeier, S. Voit, G. Carle: The Case for **Writing Network Drivers in High-Level Programming Languages,** ACM/IEEE Symposium on Architectures for Networking and Communications Systems ANCS 2019 **Best Paper Award,** Cambridge, U.K., Sept. 2019, https://www.net.in.tum.de/news/2019/ancs-best-paper-award.html

[ANCS2019b] P. Emmerich, M. Pudelko, S. Bauer, S. Huber, T. Zwickl, G. Carle: **User Space Network Drivers**

High-Level Languages for Improved Low-Level Software



Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

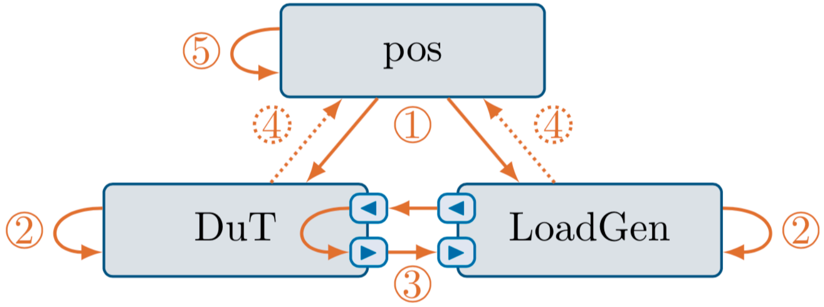
Approach

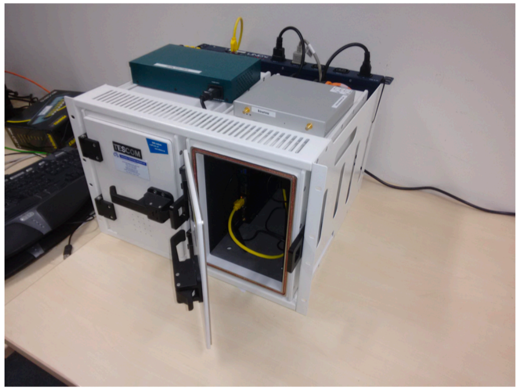
Testbeds for High-Precision Network Experiments

Testbed for Reproducible Experiments

Fully automated workflow
for reproducible network experiments

- Multi-user support
- Input: test configuration file
- Allocate resources
- Boot test machines
- Deploy system images via network
- Configure network topology
- Deploy host scripts
- Supervise test sequence
- Collect results
- Output: measurement results



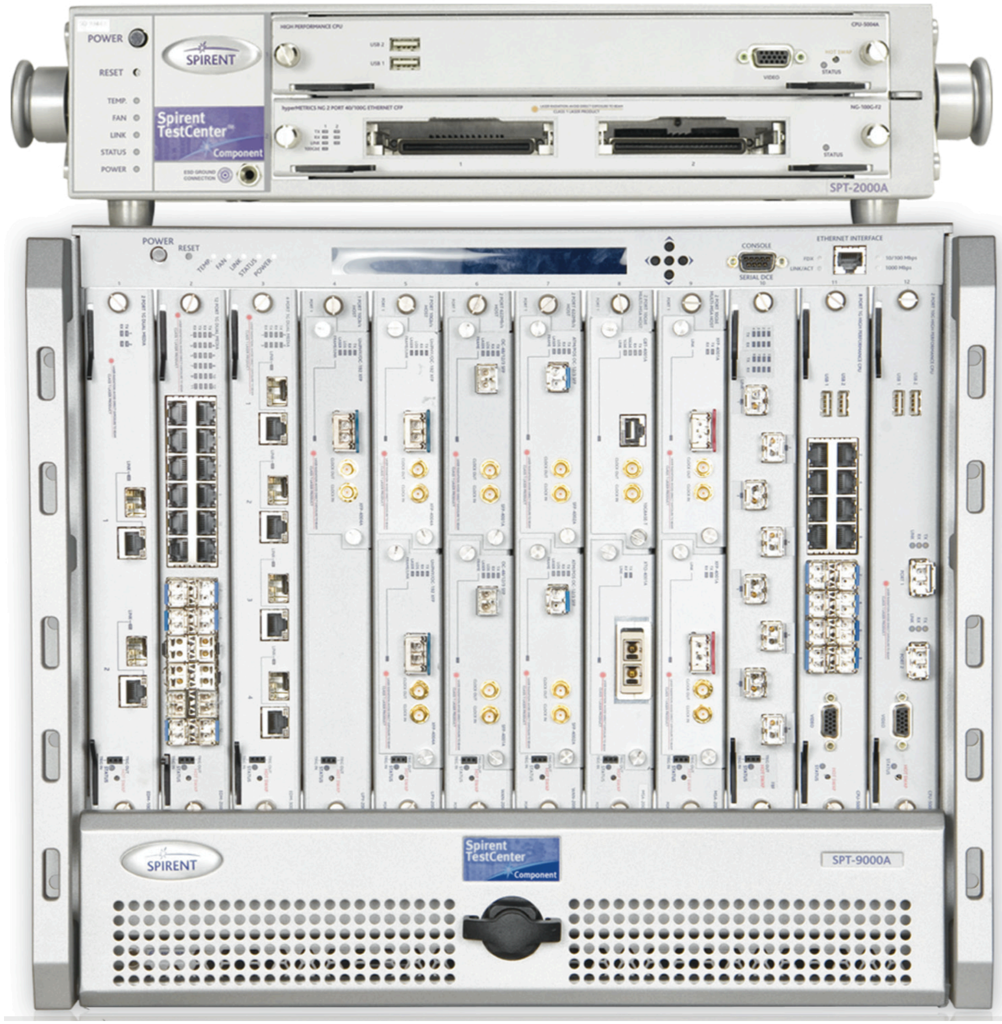


Hardware Traffic Generators

- Fast
- Precise

but

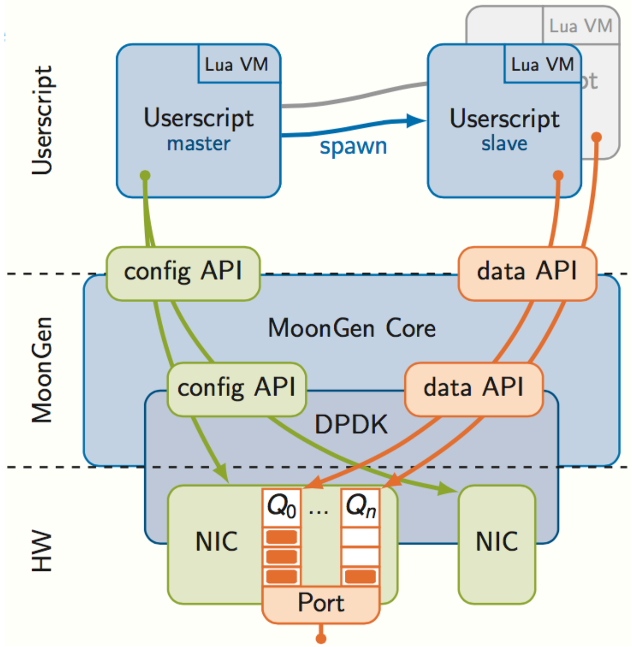
- Expensive
- Difficult to deploy
- Not flexible



Spirent traffic generator

MoonGen

- Inexpensive: Commecial Off-The-Shelf hardware
- Fast: DPDK for packet I/O, multi-core support
- Easy to deploy: simple software setup
- Flexible: user-controlled Lua scripts
- Precise
  - Timestamping: Utilize hardware features found on modern commodity NICs
  - Rate control: Hardware features and novel software approach



[IMC15] Paul Emmerich, Sebastian Gallenmüller, Daniel Raumer, Florian Wohlfart, Georg Carle: MoonGen: A Scriptable High-Speed Packet Generator, ACM Internet Measurement Conference (IMC 2015), Tokyo, Japan, October 2015

[ANRP17] Internet Research Task Force (IRTF) Applied Networking Research Prize, IETF-100, Nov. 2017, https://irtf.org/anrp

[ANCS17] Paul Emmerich, Sebastian Gallenmüller, Gianni Antichi, Andrew Moore, Georg Carle: Mind the Gap – A Comparison of Software Packet Generators, ACM/IEEE Symposium on Architectures for Networking and Communications Systems 2017

Usage of MoonGen/libmoon

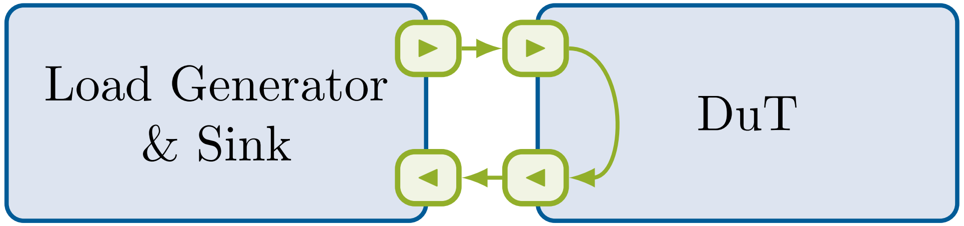
Name	Usage scenario	Publication
High-performance applications:		
FlowScope	Tool for high-performance flow capture and analysis	[11], [12]
MoonRoute	Extensible high-performance router	[4], [13]
Benchmarking tools:		
RFC 2544	Modular benchmarking tool	[14], [15]
OPNFV VSPERF	Automated NFV testing framework	[16], [17]
FLOWer	High-performance switch benchmarking	[18], [19]
Traffic & packet generation:		
NFVnice	Throughput and latency measurements	[20]
Verified NAT	Throughput and latency measurements	[21]
PISCES	Throughput measurements	[22], [23]
Sonata	Replaying CAIDA traces	[24]
DoS flood generator	DNS and TCP SYN flooding attack tools	[25]–[27]
MoonGen / libmoon under test:		
MoonGen investigation	Precise and accurate rate control and timestamping	[3], [28], [29]
MoonGen timestamping	Investigation of timestamping for packet generators	[30]
Additions to MoonGen / libmoon:		
MoonStack	Easy-to-use and efficient packet creation	[31]

[Comsnets18] Gallenmüller, Scholz, Wohlfart, Scheitle, Emmerich, Carle, “High-Performance Packet Processing and Measurements,” COMSNETS 2018, Bangalore, India, Jan. 2018

Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

Investigating Different Properties and Bottlenecks

System Analysis



Measurement setup

Black-box

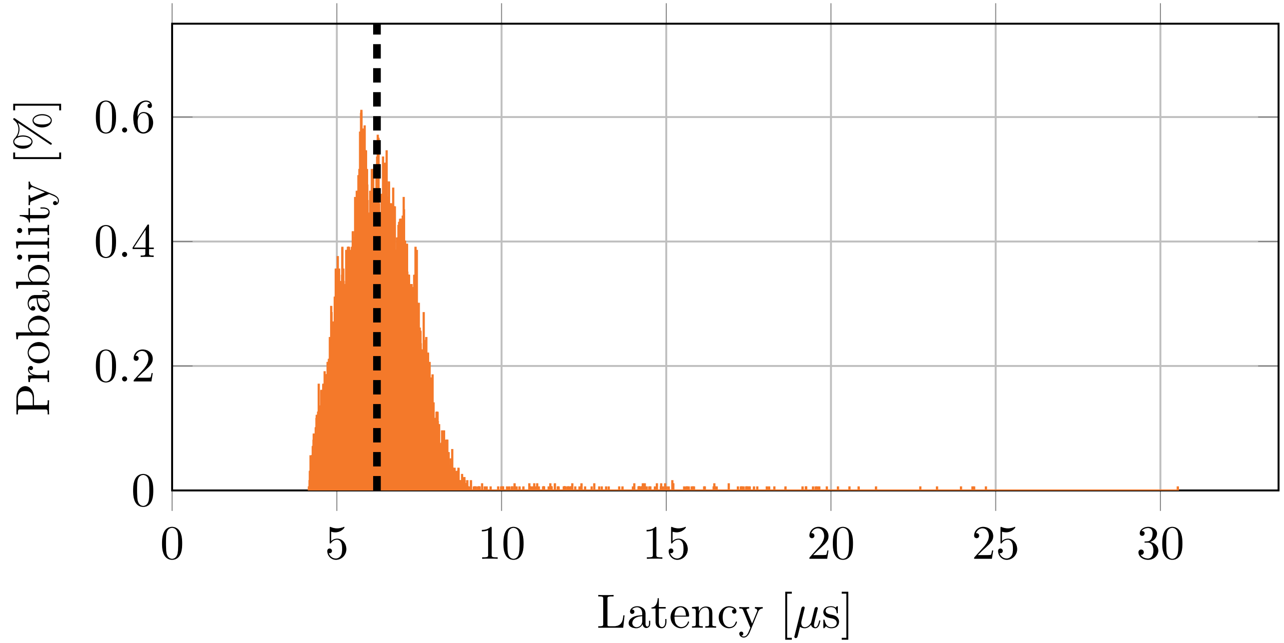
- Throughput
  - Packets per second, bytes per second
  - Frame loss rate
  - Back-to-Back frame burst size
- Latency
  - Median, average, worst case, percentiles, ...

White-box

- Hardware and software events
  - Cycles, Interrupts, L1/L2/L3 cache misses
  - Granularity: per second, per packet, per function

Latency Measurements: FreeBSD Software Router

FreeBSD router, forwarding 64-byte packets

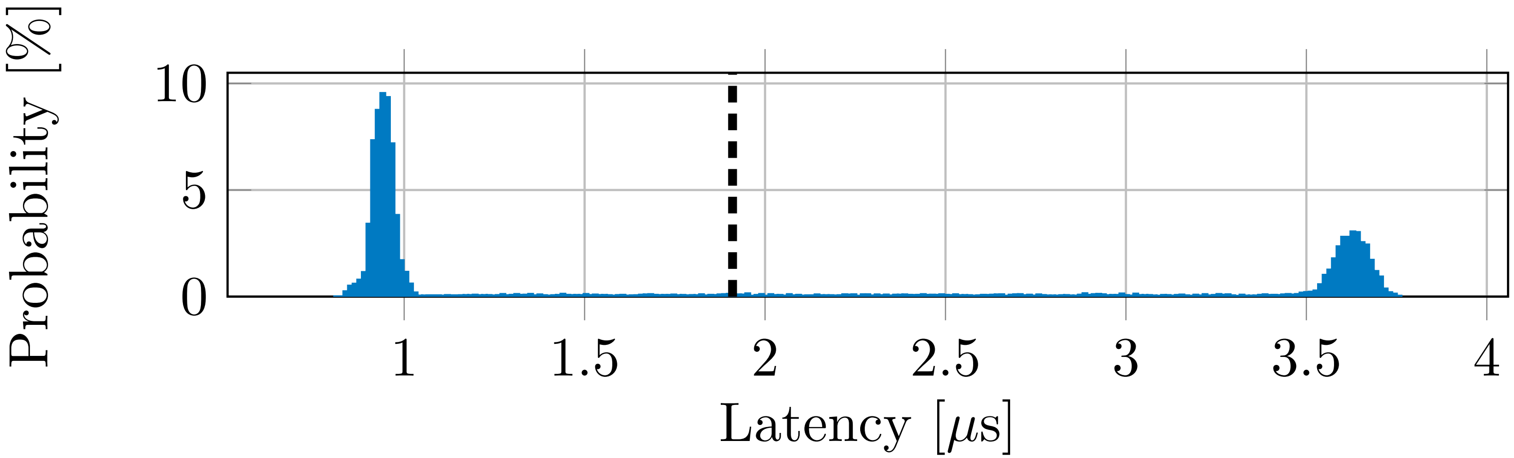


- long tail distribution

[ANRW16] Daniel Raumer, Sebastian Gallenmüller, Florian Wohlfart, Paul Emmerich, Patrick Werneck, Georg Carle: Revisiting benchmarking methodology for interconnect devices. In Applied Networking Research Workshop 2016, Jul. 2016

Latency Measurements: Pica8 Switch

Pica8 switch, forwarding 64-byte packets



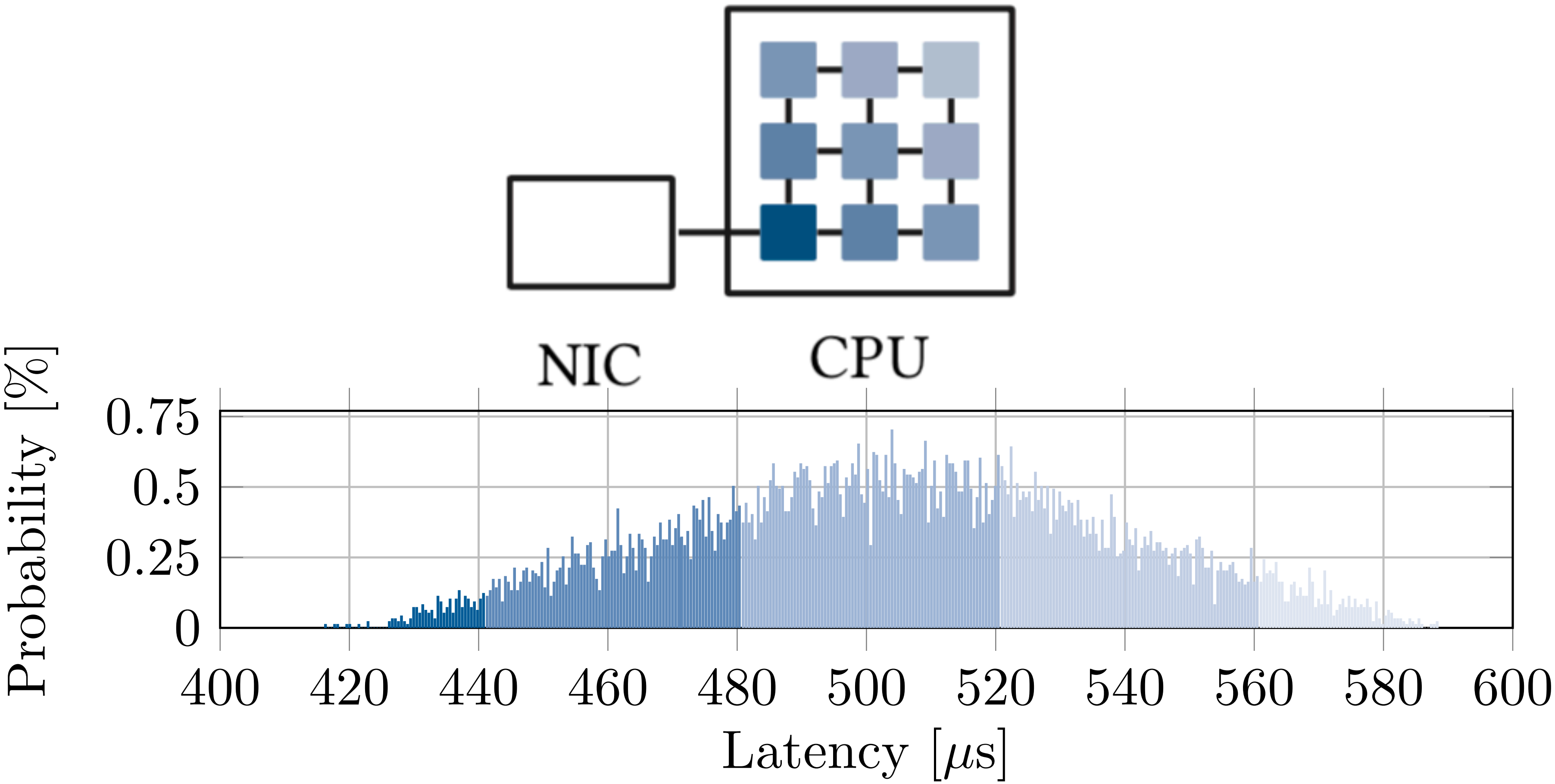
Different processing paths through device

⇨ Detailed analysis: histograms,
Short analysis: percentiles (e.g. 25th, 50th, 75th, 95th, 99th)

[IFIPNetw16] Paul Emmerich, Sebastian Gallenmüller, Georg Carle, FLOWer – Device Benchmarking Beyond 100 Gbit/s, in IFIP Networking 2016, May 2016

Explanation of latency distribution by processing paths

Mikrotik Cloud Core Router CCR1036-8G-2S+, Tilera Tile-Gx36



[ANRW16] Daniel Raumer, Sebastian Gallenmüller, Florian Wohlfart, Paul Emmerich, Patrick Werneck, Georg Carle: Revisiting benchmarking methodology for interconnect devices. Applied Networking Research Workshop 2016, Jul. 2016

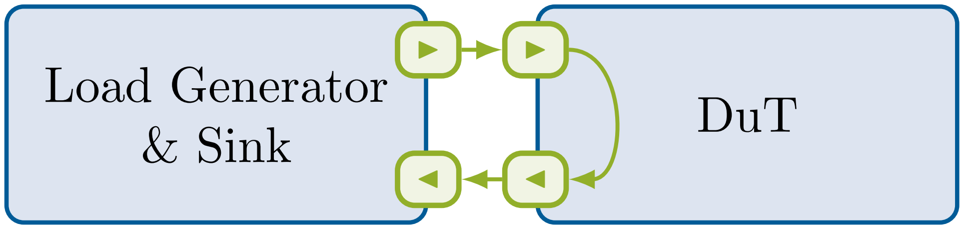
5G Low Latency Services

5G Ultra-Reliable Low-Latency Communication (URLLC)

- Ultra reliable: 99.999% packet delivery probability
- Low latency: 1ms one-way latency in Radio Access Network (RAN)

5G Service provisioning with Virtual Network Functions (VNF)

- Virtualized environment: Linux, kvm
- Network function: Snort3 (baseline setup: forwarder, no filtering)

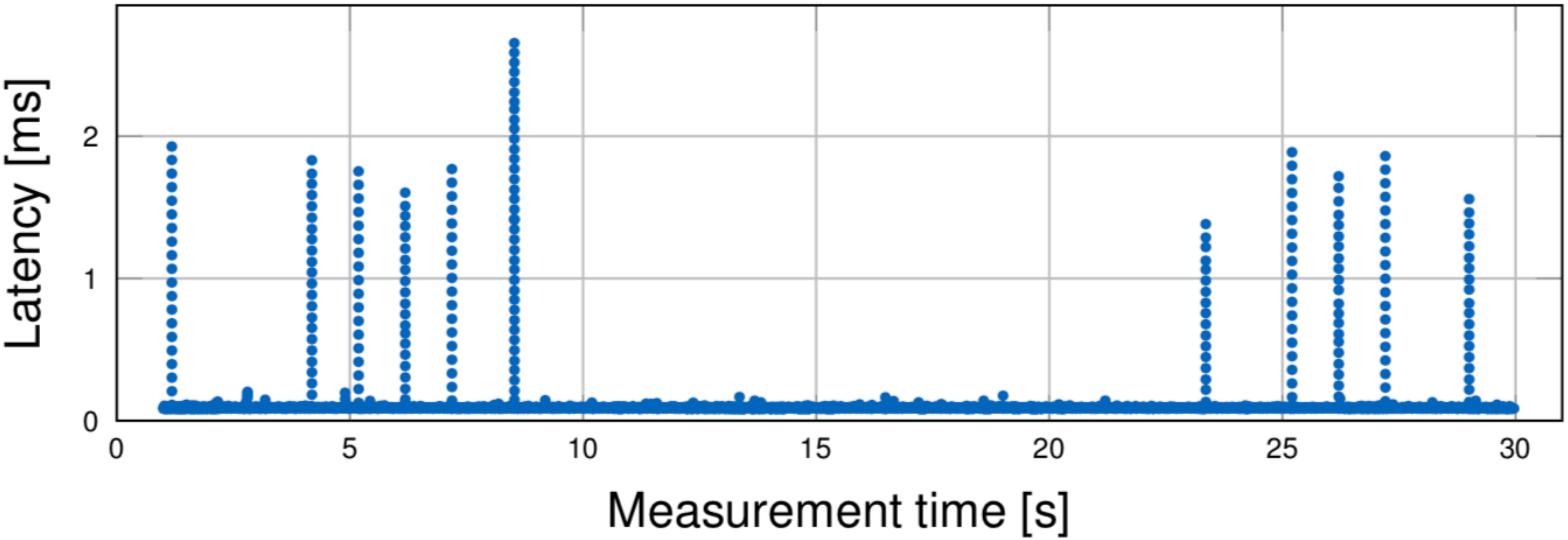


percentiles	50th	99th	99.9th	99.99th	99.999th
Snort 3 forwarder	69 µs	88 µs	107 µs	1.7 ms	2.5 ms

⇨ 99.99th percentile already violates URLLC

Latency of VNF

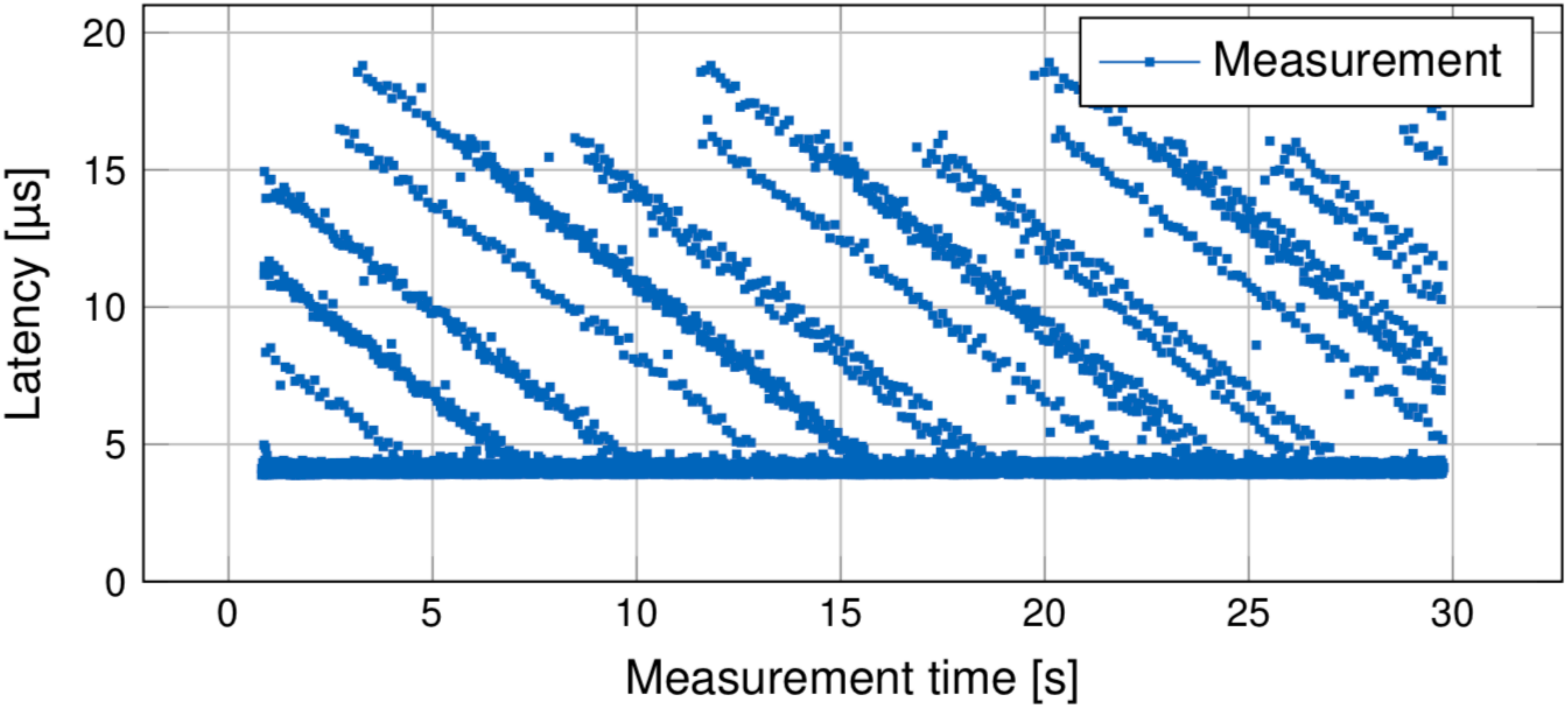
Snort 3 forwarding



[ArXiv19] Sebastian Gallenmüller, Johannes Naab, Iris Adam, Georg Carle:
5G QoS: Impact of Security Functions on Latency,
https://arxiv.org/abs/1909.08397, Nov. 2019

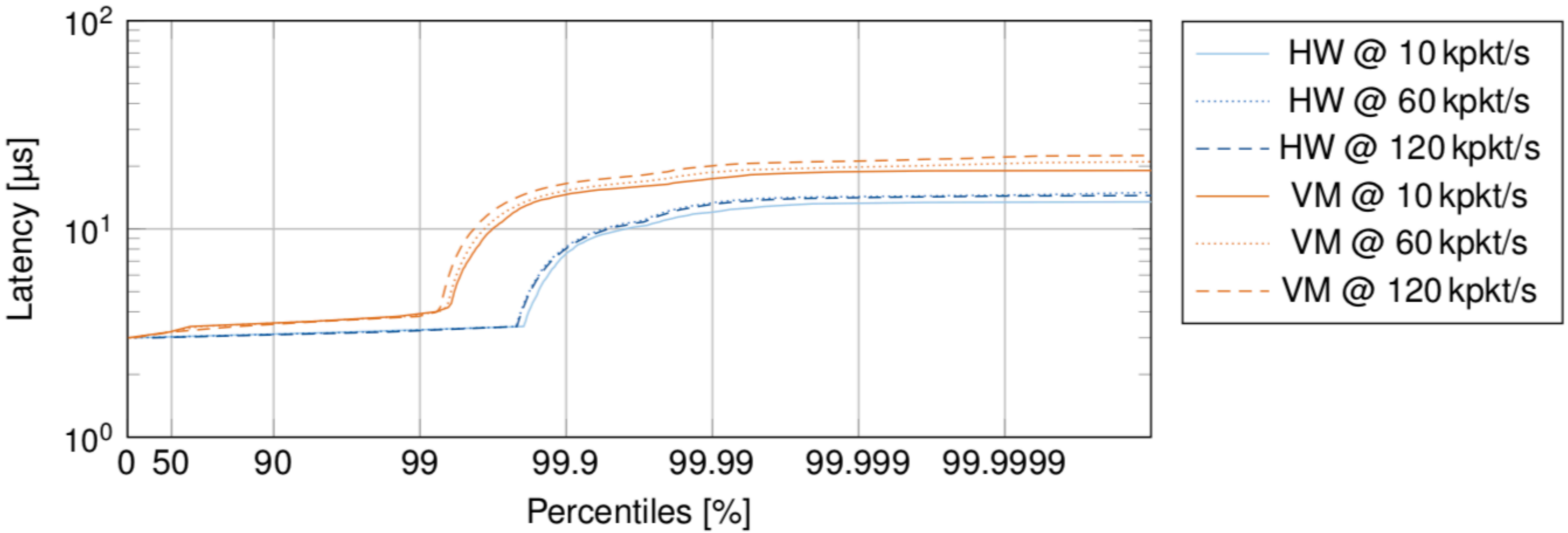
DPDK L2 Forwarding

Influence of interrupts



- 10 kpacket/s: 100 us interarrival time
- Instrumentation reveals two interrupts, rate 250 Hz and 125 Hz
  - local timer interrupts (loc): 5.5us; IRQ work interrupts (iwi): 8.2 us
- Pattern due to aliasing: interrupt duration < packet interarrival time

DPDK L2 Forwarding



- High Dynamic Range (HDR) Histogram
- DPDK L2 forwarding as baseline
- HW: no virtualisation; VM: kvm virtualisation
- Maximum latency: ~ 0,02 ms

[ArXiv19] Sebastian Gallenmüller, Johannes Naab, Iris Adam, Georg Carle: 5G QoS: Impact of Security Functions on Latency, https://arxiv.org/abs/1909.08397, Nov. 2019

Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

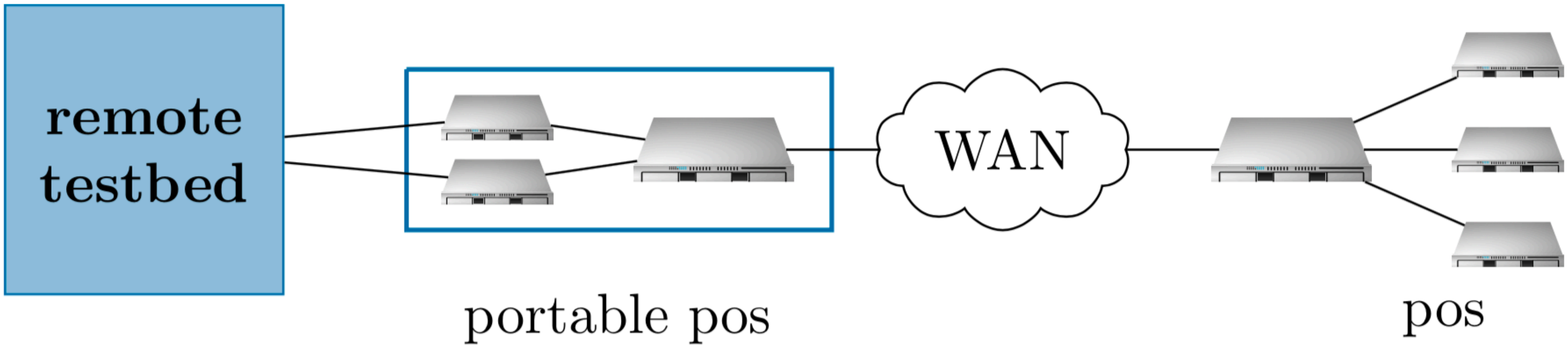
Next Steps

Federated testbeds
Virtual testbeds

Federated Testbeds

Concept for federated testbeds

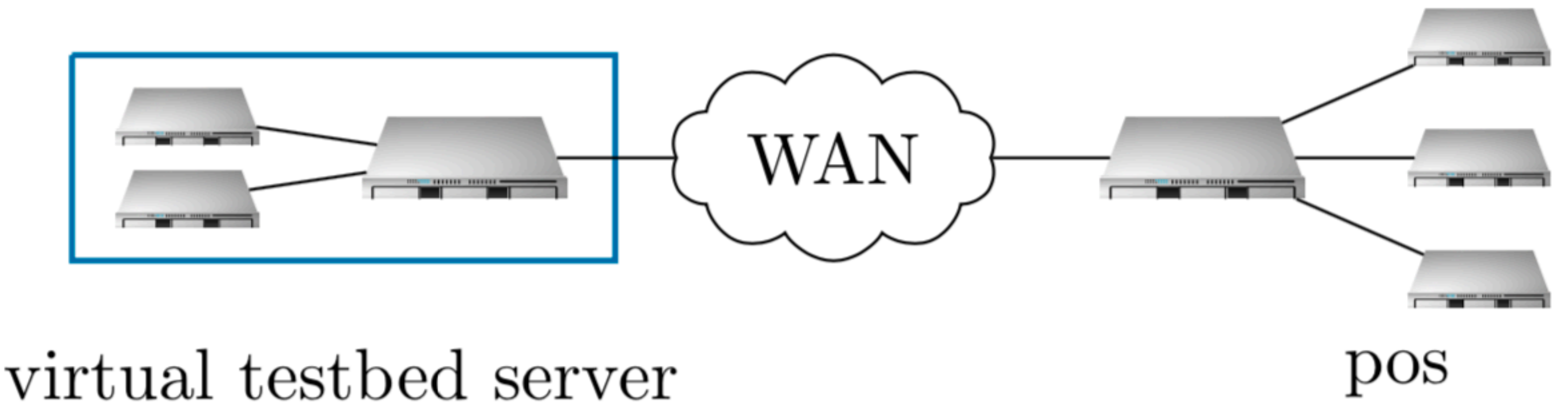
- Extend remote testbed by
- Connect experiment hosts by tunnel to TUM testbed infrastructure
- Use experiment workflow and postprocessing
- Goals
  - Equivalent experiments on different HW ⇨ transfer models
  - Link infrastructure of different technology ⇨ new experiments



Virtual Testbeds

Concept for virtual testbeds

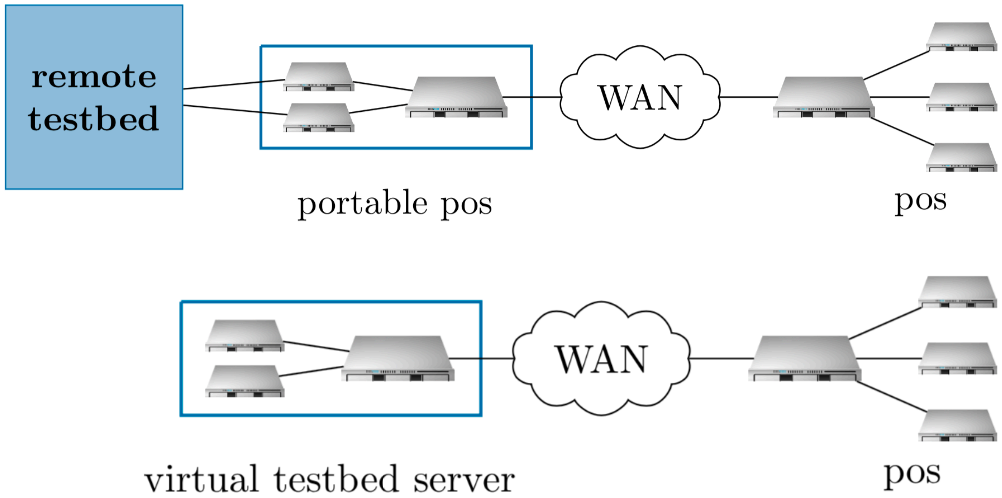
- Open-source virtual clone of a testbed
- Connect virtual testbed by tunnel to TUM testbed infrastructure
- Use experiment workflow and postprocessing
- Goal:
  - Create transfer models that describe relation between experiments conducted on real testbed and its virtual twin
  - Scale up number and quality of experiments



Reproducibility by Design - Conclusions

Challenge of reproducible networked systems experiments

- Complex hardware + software architectures
- Data-driven research for understanding root causes
- High data quality: Precision, accuracy, coverage
- Traffic generation and measurements: COTS HW, flexibility
- TUM testbed infrastructure and tools for reproducible experiments
- Next steps
  - Federation of testbeds
  - Virtual testbeds



Questions?