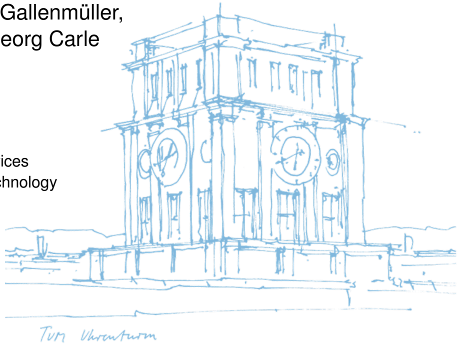
Hardware-assisted Virtual Networking for low-latency network services

Florian Wiedner, Max Helm, Sebastian Gallenmüller,
Alexander Daichendt, Jonas Andre, Georg Carle

Tuesday 13th May, 2025

Chair of Network Architectures and Services
School of Computation, Information, and Technology
Technical University of Munich



Motivation

Latency of a Virtual Network Function Chain: Operating on Low-Latency

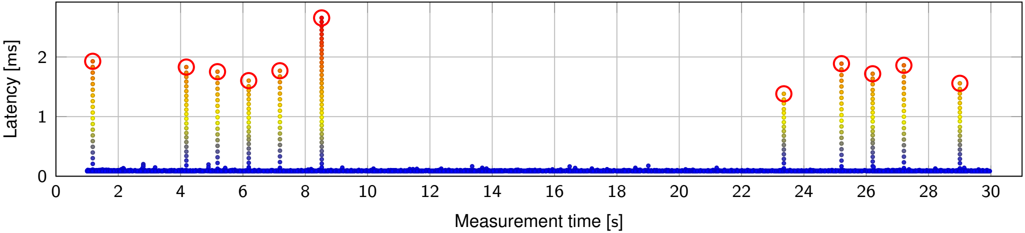


Figure: Snort 3 forwarder worst-case latencies (single-node)

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

- Ultra reliable: 99.999% success probability
- Low latency: 1 ms one-way end-to-end latency in the radio access network (RAN)[1]

URRLC violations happen irregularly over the entire measurement

[1] ITU, Report ITU-R M.2410-0 (11/2017) Minimum requirements related to technical performance for IMT-2020 radio interface(s), https://www.itu.int/dms_pub/itu-r/opb/rep/R-REP-M.2410-2017-PDF-E.pdf, Accessed: 2025-05-05.

Goal

Enabling ultra low-latency in general-purpose networks

This requires

- Sharing the network between multiple customers and service level requirements
- Resource sharing and on-demand provisioning of resources

Goal

Enabling ultra low-latency in general-purpose networks

This requires

- Sharing the network between multiple customers and service level requirements
- Resource sharing and on-demand provisioning of resources

We show that

- Low-latency with virtualizations on commodity hardware is possible
- Using careful planning and optimization, different virtualization solutions can be used.

Goal

Enabling ultra low-latency in general-purpose networks

This requires

- Sharing the network between multiple customers and service level requirements
- Resource sharing and on-demand provisioning of resources

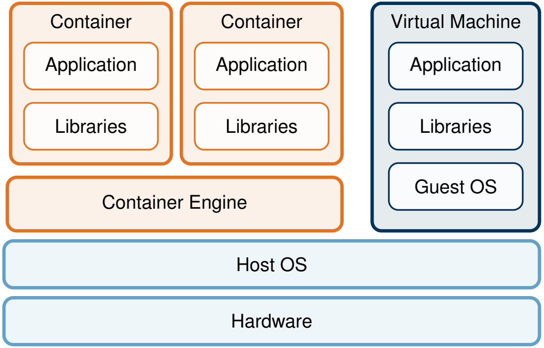
We show that

- Low-latency with virtualizations on commodity hardware is possible
- Using careful planning and optimization, different virtualization solutions can be used.

We will talk about

- **Optimizations**: How can VMs and containers be optimized towards low-latency
- **Measurement Setup**: What is needed to make the data comparable
- **Evaluation**: What our experiments did show
- Our **Recommendations**

Virtualization: Virtual Machines vs. Containers



Containers (e.g. LXC)

- Lightweight OS-level virtualization
- Shared kernel
- Isolated applications

Virtual Machines (VMs) (e.g. KVM)

- Full OS virtualization
- No shared kernel
- Isolated OS

Low-Latency Optimizations

Challenges & Solutions

Reasons for virtualization latency performance impairment

- Interupt-based IO
  - Linux NAPI
- CPU features
  - Dynamic scheduling of processes onto CPU cores
  - Virtual cores (SMT)
  - Energy-saving mechanisms
- Expensive VM IO

Low-Latency Optimizations

Challenges & Solutions

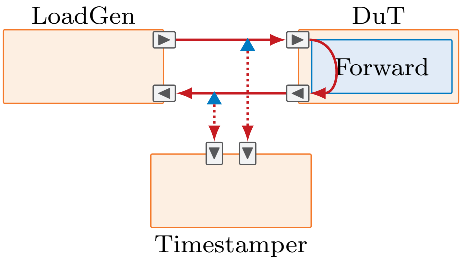
Reasons for virtualization latency performance impairment

- Interupt-based IO
  - Linux NAPI
- CPU features
  - Dynamic scheduling of processes onto CPU cores
  - Virtual cores (SMT)
  - Energy-saving mechanisms
- Expensive VM IO

Fixing Virtualization performance

- Polling-based IO
  - DPDK
- CPU features
  - Statically allocate CPU cores for processes
  - Disable SMT
  - Disable energy-saving mechanisms
- SR-IOV

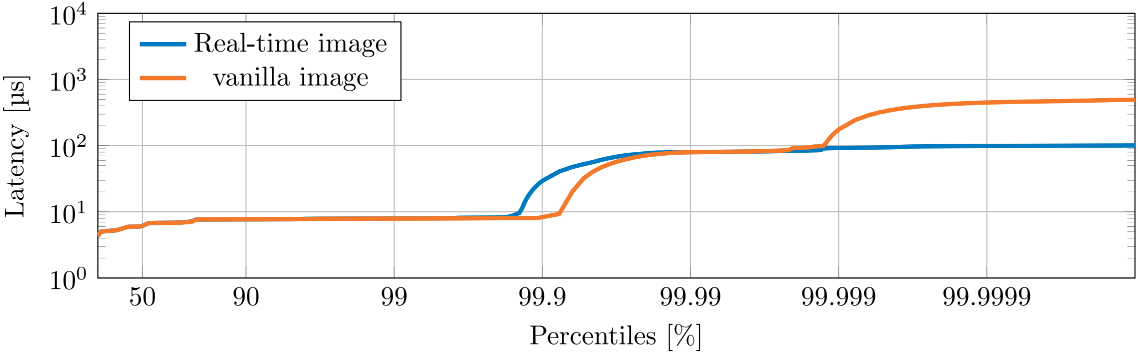
Measurement Setup



- Loadgen runs a packet generator (MoonGen) creating UDP packets
- Device under Test (DuT) runs containers/VMs/packet processing application
- Timestamper records DuT ingress/egress traffic (passive optical Terminal Access Points)
    - Hardware-timestamping of entire network traffic (timer resolution 1.25 ns)
    - Determine worst-case latencies
- Traffic: UDP Traffic with 64 B packets
- Duration: 160 s per measurement

Evaluation

Kernel Variants on LXC Container [1 Mpackets/s]

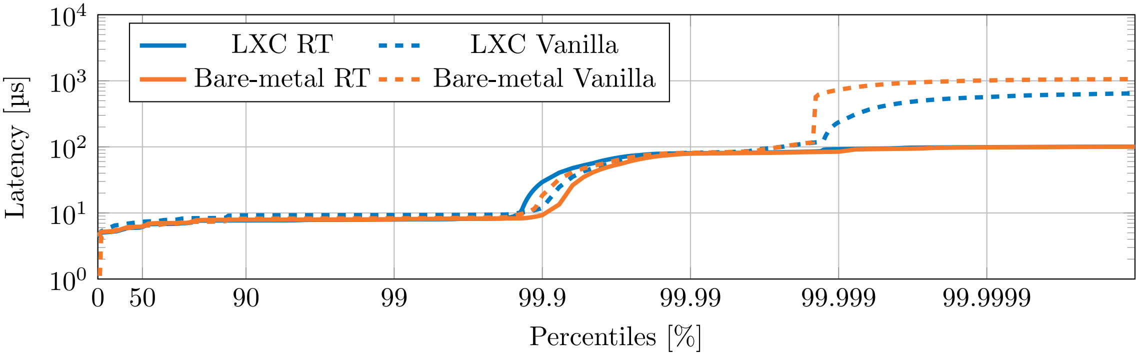


The kernel variant is significantly influencing the tail-latency.

- Nearly similar results until 99.999*th* percentile
- Real-time kernel performs most deterministic

Evaluation

Containers vs. Bare-metal [1 Mpackets/s]

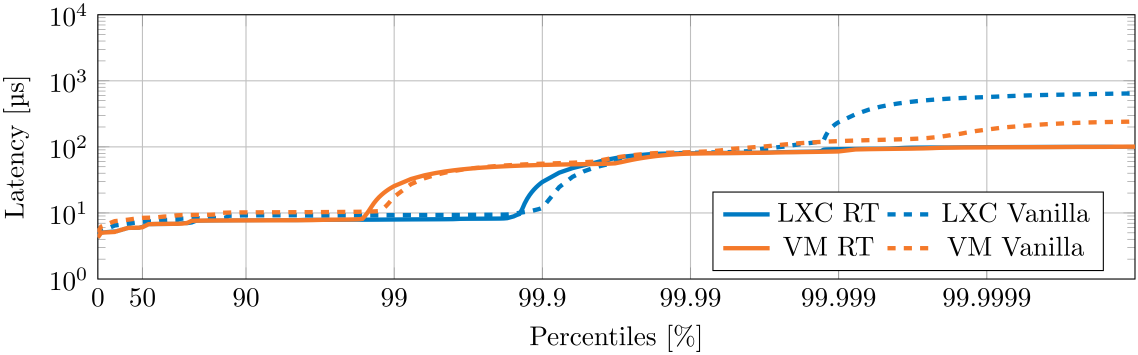


Bare-metal not significantly lower tail-latency

- Non-optimized version outperforms LXC bare-metal due to minimal isolation
- No significant difference when optimized

Evaluation

Containers vs. VMs [1 Mpackets/s]

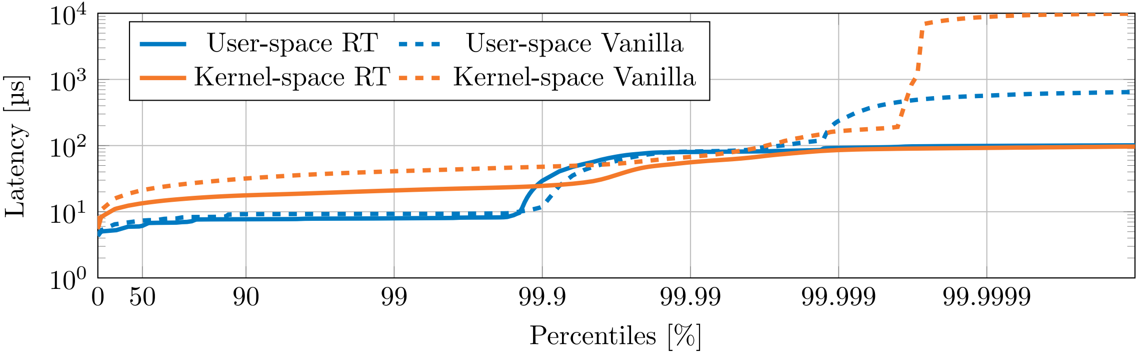


VMs not significantly lower tail-latency

- Non-optimized version outperforms VMs LXC due to higher degree of isolation
- No significant difference when optimized

Evaluation

Kernel vs. User-space [1 Mpackets/s]



Optimized kernel-space networking similar to user-space

- In non-optimized system user-space networking clearly outperforms kernel-space
- No significant difference when optimized

Publications and Measurement Data

Content is based on our publications

Container F. Wiedner, M. Helm, A. Daichendt u. a., „Containing Low Tail-Latencies in Packet Processing Using Lightweight Virtualization,“ in 2023 35rd International Teletraffic Congress (ITC-35), Turin, Italy, Okt. 2023

F. Wiedner, M. Helm, A. Daichendt u. a., „Performance evaluation of containers for low-latency packet processing in virtualized network environments,“ Performance Evaluation, Jg. 166, S. 102 442, 2024, ISSN: 0166-5316

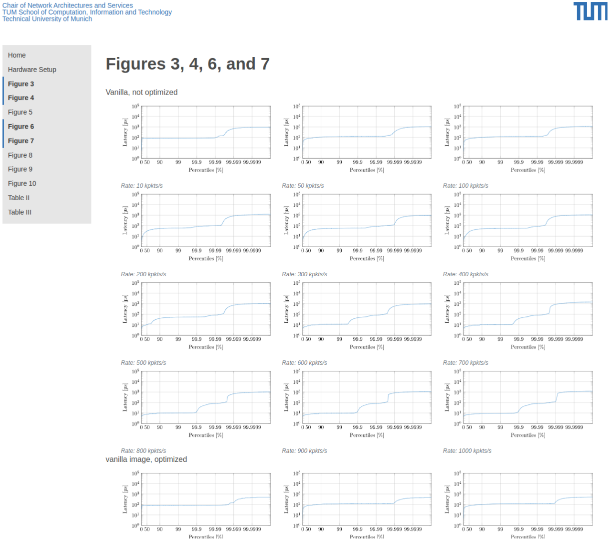
Topologies F. Wiedner, M. Helm, S. Gallenmüller u. a., „HVNet: Hardware-Assisted Virtual Networking on a Single Physical Host,“ in IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops, IEEE, 2022, S. 1–6

Baremetal S. Gallenmüller, F. Wiedner, J. Naab u. a., „How Low Can You Go? A Limbo Dance for Low-Latency Network Functions,“ J. Netw. Syst. Manag., Jg. 31, Nr. 1, S. 20, 2023

VMs S. Gallenmüller, F. Wiedner, J. Naab u. a., „Ducked Tails: Trimming the Tail Latency of(f) Packet Processing Systems,“ in 17th International Conference on Network and Service Management, CNSM 2021, Izmir, Turkey, October 25-29, 2021, P. Chemouil, M. Ulema, S. Clayman u. a., Hrsg., IEEE, 2021, S. 537–543

Publications and Measurement Data

- Available artifacts:
  - Evaluation scripts
  - Measurement data
  - Reproducibility
- Website for container and comparison: `https://wiednerf.github.io/containerized-`
- Website for Motivation and Bare-metal Optimizations: `https://gallenmu.github.io/latency-limbo/`
- Website for Virtual Machine Optimizations: `https://gallenmu.github.io/hipnet21/`
- → Allow to utilize results in network validation, production, and research



Low Tail-Latencies in Packet Processing Systems with Virtualization

- Similar tail-latencies between container, bare-metal, and VMs
- More influence of shared OS in lightweight systems
- Latency excluded as primary selection criterion for technology
- Low-latency systems demand more resources than unoptimized ones
- Topology evaluation as example application

Low Tail-Latencies in Packet Processing Systems with Virtualization

- Similar tail-latencies between container, bare-metal, and VMs
- More influence of shared OS in lightweight systems
- Latency excluded as primary selection criterion for technology
- Low-latency systems demand more resources than unoptimized ones
- Topology evaluation as example application

Recommendations

Technology	Latency	Security	Resources
VM	×	○	○
Container	○	×	✓
Bare-metal	✓	✓	×

Further information:

Papers are available online.

Container

Optimizations

Bare-metal

Bibliography

[1] ITU, Report ITU-R M.2410-0 (11/2017) Minimum requirements related to technical performance for IMT-2020 radio interface(s), `https://www.itu.int/dms_pub/itu-r/opb/rep/R-REP-M.2410-2017-PDF-E.pdf`, Accessed: 2025-05-05.

[2] F. Wiedner, M. Helm, A. Daichendt, J. Andre, and G. Carle, „Containing Low Tail-Latencies in Packet Processing Using Lightweight Virtualization“, in 2023 35rd International Teletraffic Congress (ITC-35), Turin, Italy, Oct. 2023.

[3] F. Wiedner, M. Helm, A. Daichendt, J. Andre, and G. Carle, „Performance evaluation of containers for low-latency packet processing in virtualized network environments“, Performance Evaluation, vol. 166, p. 102 442, 2024, ISSN: 0166-5316.

[4] F. Wiedner, M. Helm, S. Gallenmüller, and G. Carle, „Hvnet: Hardware-assisted virtual networking on a single physical host“, in IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops, IEEE, 2022, pp. 1–6.

[5] S. Gallenmüller, F. Wiedner, J. Naab, and G. Carle, „How low can you go? A limbo dance for low-latency network functions“, J. Netw. Syst. Manag., vol. 31, no. 1, p. 20, 2023.

[6] S. Gallenmüller, F. Wiedner, J. Naab, and G. Carle, „Ducked tails: Trimming the tail latency of(f) packet processing systems“, in 17th International Conference on Network and Service Management, CNSM 2021, Izmir, Turkey, October 25-29, 2021, P. Chemouil, M. Ulema, S. Clayman, M. Sayit, C. Çetinkaya, and S. Secci, Eds., IEEE, 2021, pp. 537–543.

Backup: Application

Full Network-Domain Application

Emulation of L3 Network

- Utilize typical domain technologies: OSPF
- Software Router based on FRRouting
- Analyzing influence on latency with link failures

$\rightarrow$ Hardware-supported Virtualization enables full network application evaluation in latency-aware scenarios

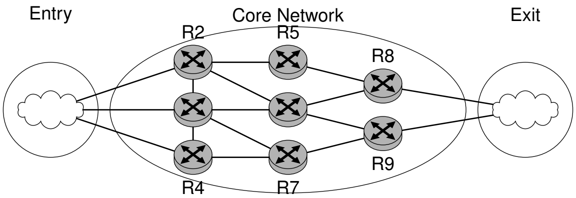
Backup: Application

Full Network-Domain Application

Emulation of L3 Network

- Utilize typical domain technologies: OSPF
- Software Router based on FRRouting
- Analyzing influence on latency with link failures

→ Hardware-supported Virtualization enables full network application evaluation in latency-aware scenarios



Topology with multiple same-hop-length paths

Backup: Application

Topology Adoption due to Link Failure

Experiment Settings

- four flows from entry to exit node
- Same setup as prior measurements
- Each flow with 10 Mbit/s and 363 B packet size

Backup: Application

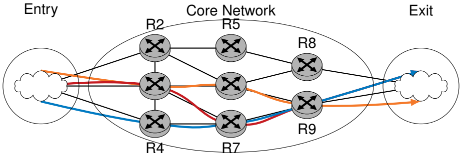
Topology Adoption due to Link Failure

Experiment Settings

- four flows from entry to exit node
- Same setup as prior measurements
- Each flow with 10 Mbit/s and 363 B packet size

Experiment Plan

- Two Links are unavailable after 30 s
- OSPF need to react on network changes
- → What influences are visible in the network data and per-flow end-to-end-latency?



→ Initial routing with four flows using three paths

Backup: Application

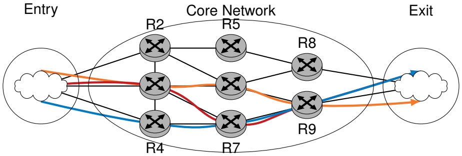
Topology Adoption due to Link Failure

Experiment Settings

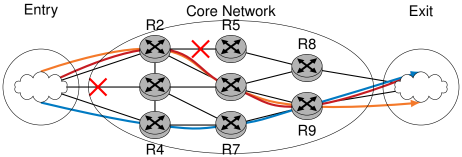
- four flows from entry to exit node
- Same setup as prior measurements
- Each flow with 10 Mbit/s and 363 B packet size

Experiment Plan

- Two Links are unavailable after 30 s
- OSPF need to react on network changes
- → What influences are visible in the network data and per-flow end-to-end-latency?



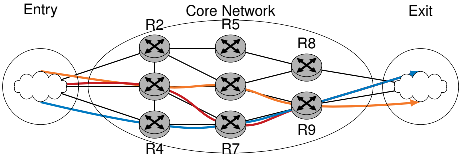
→ Initial routing with four flows using three paths



→ Rerouting due link failure after 30 s

Backup: Application

Evaluation



→ Initial routing with four flows using three paths

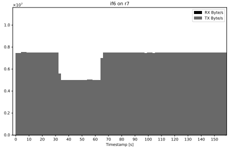


Figure 1: Interface between R7 and R9

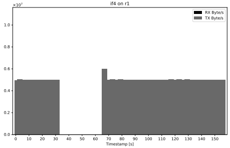
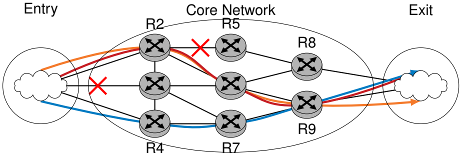


Figure 3: Interface between R1 and R3



→ Rerouting due link failure after 30 s

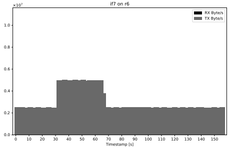


Figure 2: Interface between R6 and R9

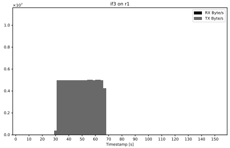
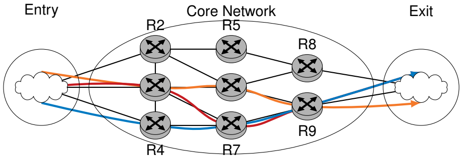


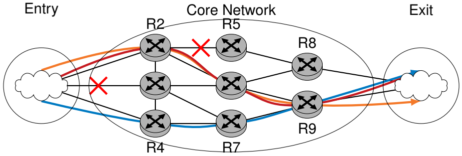
Figure 4: Interface between R1 and R2

Backup: Application

Evaluation



→ Initial routing with four flows using three paths



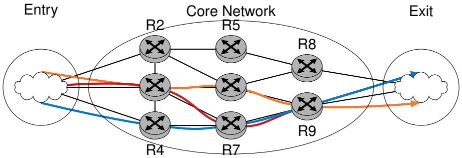
→ Rerouting due link failure after 30 s

Latency (average)

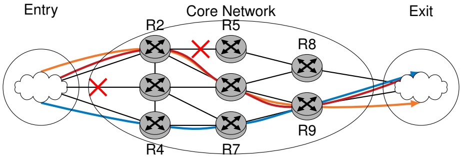
- Initially: between 288 µs and 295 µs

Backup: Application

Evaluation



→ Initial routing with four flows using three paths



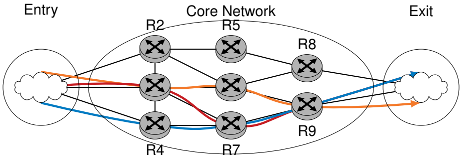
→ Rerouting due link failure after 30 s

Latency (average)

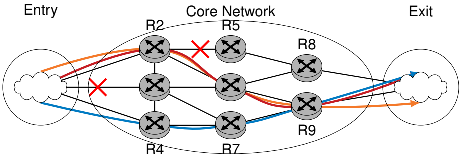
- Initially: between 288 µs and 295 µs
- During Link failure: between 294 µs and 299 µs

Backup: Application

Evaluation



→ Initial routing with four flows using three paths



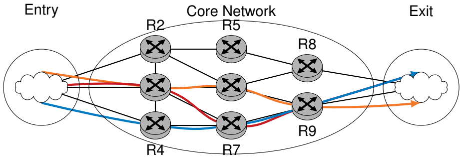
→ Rerouting due link failure after 30 s

Latency (average)

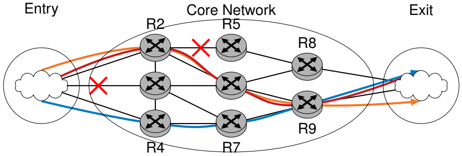
- Initially: between 288 µs and 295 µs
- During Link failure: between 294 µs and 299 µs
- → routing protocol-based low latency experiments possible

Backup: Application

Evaluation



→ Initial routing with four flows using three paths



→ Rerouting due link failure after 30 s

Latency (average)

- Initially: between 288 µs and 295 µs
- During Link failure: between 294 µs and 299 µs
- → routing protocol-based low latency experiments possible

Evaluation of the Experiment

- Experiment shows the potential on using virtualization for networking
- Potential use-cases are flexible validations or network functions
- Careful optimizing the virtualized environment allow sophisticated experiments and applications