Lehrstuhl für Netzarchitekturen und Netzdienste
Fakultät für Informatik
Technische Universität München

**Antrittsvortrag zur Bachelorarbeit**

# Performance-Analyse von VPP

| | |
|---|---|
| Name: | Peter **Okelmann** |
| Betreuer: | Paul Emmerich, Dominik Scholz |
| Aufgabensteller: | Prof. Dr.-Ing. Georg Carle |
| Beginn: | 12/2018 |
| Ende: | 04/2019 |

## Topic

VPP (Vector Packet Processing) [1] is an open source software for providing efficient network switching and routing. In contrary to common hardware switches, the whole packet processing is done in software. This allows a more versatile use of the same router in different applications and provides flexibility regarding hardware it can run on.

Typically hardware routers are expected to be faster than software routers, but VPP has several approaches to perform better compared to similar software projects like Open vSwitch [2] or the Linux Router: As the name indicates, it processes packages in vectors, in other words, multiple packages at a time which reduces overhead per package. Furthermore it doesn't use the slow linux network drivers for it's interfaces, but an own one.

The scope of this Bachelor's thesis shall be to measure the performance of VPP and to evaluate which scenarios are important to be tested.

## Approach

For this Bachelor's Thesis automated tests shall be implemented to run test scenarios on the Baltikum testbed. Moongen [3] shall be used on the tester to generate loads.

VPP shall be tested torwards IPv4 versus IPv6, packet sizes, cpu scaling (optionally multi socket NUMA architectures) and routing table sizes (BGP table size $>$600.000 entries in 2017 [4]). Optionally one specific VPP feature can be analyzed, too, like tunneling, firewall or NAT.

Measurement results shall contain package count (per time), latency, package loss and whitebox testing results: Cpu load and cache misses.

In the end a conclusion shall be made, giving VPP a rough rank relative to real world routing solutions.

** optional: VPP performance in quemu

** important related work: [5] [6] [7] [8]

## Timetable

| Deadline after *n* Months | Task |
|---|---|
| 0.5 | automate a basic testbench setup and run a test using moongen for the tester and VPP for the DUT |
| 1.0 | run tests with changing parameters |
| 1.5 | create result visualization pilelines and analyze them |
| 2.5 | improve tests and extend them by whitebox testing |
| 3.0 | analyze new results and review testing methodology |
| 4.0 | Assemble Thesis from analysiss and testing results |

# Literatur

[1] FD.io, “Vpp website,” https://wiki.fd.io/view/VPP, accessed on 2018-11-10.

[2] https://www.openvswitch.org/.

[3] https://github.com/emmericp/MoonGen.

[4] http://www.bgphelp.com/2017/01/01/bgpsize/.

[5] https://wiki.fd.io/view/VPP/What_is_VPP%3F#Performance_Expectations.

[6] https://docs.fd.io/csit/master/trending/introduction/index.html.

[7] D. Raumer, S. Gallemüller, F. Wohlfart, P. Emmerich, Patrick, , and G. Carle, “Revisiting benchmarking methodology for interconnect devices,” 2016.

[8] R. V. Rosa and C. E. Rothenberg, “Taking open vswitch to the gym: An automated benchmarking approach,” 2017.