**HVNet**

Hardware-assisted Virtual Netw0rking for low-latency network services

**Motivation**

Tail-latency Experiments on Real Hardware
- Expensive
- Restrictions on creating arbitrary topologies
- Limited availability
- Can have long delivery times
- Difficult to reproduce

Tail-latency Experiments on Simulation, e.g., OMNeT++ [5]
- Restricted in terms of usable software
- No access to hardware-level features

**Background**

Single-Root I/O-Virtualization (SR-IOV):
- Share hardware resources among VMs
- Physical Function (PF) with full control
- Lightweight Virtual Function (VF) with restricted access
- Each physical device: 1 PF, multiple VFs
- Allow direct passthrough of VFs due to be PCIe-functions

Virtualization of systems is possible using:
- Full virtualization, e.g., virtual machines (VMs)
- OS-level virtualization, e.g., container

Computed raise of latency is according to [2] mostly caused by:
- Interrupts raised on the same core as the virtualization
- Energy-saving mechanism during idle times
- Other applications running on the same cores as the virtualization

**Measurement Setup**

- LoadGen runs a packet generator (MoonGen [1] creating UDP packets
- Device under Test (DuT) contains to be analyzed system
- Timestamper records ingress/egress traffic using passive optical traffic access points (TAPs)
  - Hardware-timestamping of entire network traffic (resolution 12.5 ms)
  - Determine worst-case latencies on a per-flow basis

**Area A: Tail-latency measurements [3]**

Bare-metal vs SR-IOV on VM using DPDK i2-forwarder
- Using optimizations such as to:
  - reduce interrupts
  - reduce timer-ticks
  - disable energy-saving-mechanism
  - use SR-IOV for sharing networking hardware access
- Latency between 3 and 4 µs, almost no difference between HW and VM
- Higher rates result in latency increase on a higher percentile

**Area B: HVNet - Virtualization of topologies [7]**

- One VLAN-ID per connection
- Split NIC into VFs
- Each packet traverses the wire per link once
- Reduce impact using optimization such as core isolation [3]

**Area C: Network services and Flow measurements**

HVNet vs. Mininet using 1 Mbit/s Flows:

Further, we use HVNet to analyze network services such as:
- QoS-aware routing algorithms in different scenarios [6]
- TSN and TSN asynchronous traffic shaping algorithms
- and use the data to predict flow-behavior, e.g., Helm et al [4].

---