Linux auf dem Switch

Benedikt “Katze” Brenner

Menzel IT GmbH

Intro

Ich

- GNU/Linux seit >10 Jahren
- Dev[1] & Ops
  - dev: Rust, C, Shell
  - sysadmin: GNU/Linux, Container, Puppet, Ansible
- Netzwerk: IPv6, Routing

[1] github.com/benaryorg

HPC

- High Performance Computing
  - viel Leistung
  - wenig Abstraktion

Menzel IT

- Expertise

Menzel IT

- Expertise
  - Serverhardware

Menzel IT

- Expertise
  - Serverhardware
  - Storage

Menzel IT

- Expertise
  - Serverhardware
  - Storage
  - Netzwerk

Menzel IT

- Expertise
  - Serverhardware
  - Storage
  - Netzwerk
- Beratung

Menzel IT

- Expertise
  - Serverhardware
  - Storage
  - Netzwerk
- Beratung
- Automatisierung

Menzel IT

- Expertise
  - Serverhardware
  - Storage
  - Netzwerk
- Beratung
- Automatisierung
- Schulung

Talk

- Switchsoftware: Überblick
  - Vor- und Nachteile/Caveats
- Switch Betriebsysteme
  - Details zum Unterbau

Disclaimer

- beruflich enge Kooperation mit Mellanox (nun Nvidia)
- SysAdmin Background
- HPC Bereich

Switches

Definition – Switch

- mehr als 2 Ports
- Link-Layer (?)
- Hardware für Offloading

Definition – Router

- low-end Router oftmals ohne Switch-Integration
- Namen sind Schall und Rauch
- “Warum kein OpenWRT?”
  - Performance
  - Hardware Integration[2]

[2] twitter.com/Toble_Miner/status/1089053538330779648

Definition – “Managed”

- Internet-Layer
  - mindestens für die Oberfläche
- Features
  - LACP
  - VLAN
  - RSTP

Beispiele

- Unmanaged
- Managed
  - QNAP QSW-M408-2C
  - TP-Link TL-SG3428X
  - Supermicro SSE-G3648B
  - MikroTik CRS305-1G-4S+IN
  - MikroTik CRS326-24S+2Q+RM
  - Mellanox/Nvidia SN2010

Unmanaged

- Ethernet
- Link-Layer von einem zum nächsten Port

Abbildung 1: Unmanaged Switch

QNAP QSW-M408-2C

- 4×10G (RJ-45/SFP+)
- übliche Managed Features

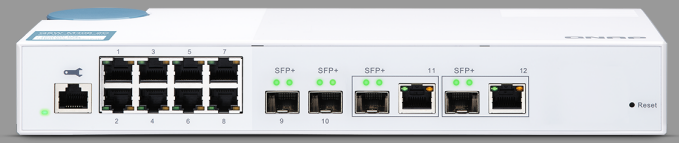

Abbildung 2: QNAP QSW-M408-2C

TP-Link TL-SG3428X

- 4×10G (SFP+)
- *viele* Managed Features

Abbildung 3: TP-Link TL-SG3428X

Supermicro SSE-G3648B

- 4×10G (SFP+)
- ~~kaputte~~ *alle* Managed Features[a]
- USB (!) ⇒ später mehr

[a] vielleicht, später mehr dazu

Abbildung 4: Supermicro SSE-G3648B

MikroTik CRS305-1G-4S+IN

- 4×10G (SFP+)
- *fancy* Managed Features

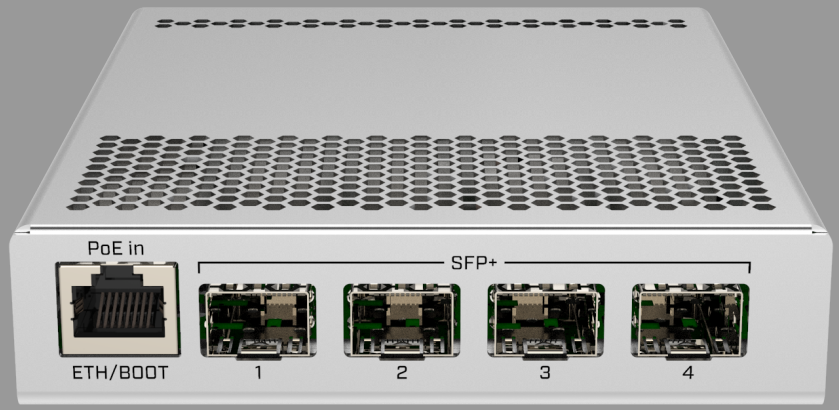

Abbildung 5: MikroTik CRS305-1G-4S+IN

MikroTik CRS326-24S+2Q+RM

- 24×10G (SFP+)
- 2×40G (QSFP+)
- *fancy* Managed Features
- USB (!) ⇒ später mehr

Abbildung 6: MikroTik CRS326-26S+2Q+RM

Nvidia SN2010

- 18×25G (SFP28)
- 4×100G (QSFP28)
- “alle” Managed Features
- USB (!) ⇒ später mehr

Abbildung 7: Nvidia SN2010

Feature-Sets

Reminder

Angefangen bei den proprietären Systemen, basierend auf GNU/Linux, die Ansätze von Docker/OCI, offenen Switch-Distributionen bis hin zu nativen Linux-Treibern hat sich in den letzten Jahren viel im Netzwerkumfeld getan. DPDK, switchdev, SONiC, Cumulus, Docker/OCI auf Switches, ONIE. Wir schauen uns einmal an, [...]

Beschreibung des Talks, Katze, 2023

Proprietäre Systeme

- können Switching
- herstellerspezifische Bezeichnungen für gleiche Features
- oft mäßiger Support für Standards
  - "wir haben da mal was eigenes dazu gemacht"
  - "jetzt muss das aber noch irgendwie kompatibel bleiben"
- oft eigene Standards
  - *MLAG*[3]
- Cisco-esque CLI und/oder Webinterface

[3] github.com/netbox-community/netbox/issues/2830

Proprietäre (GNU/)Linux Systeme

- Beispiele: MikroTik RouterOS, Nvidia ONYX, QNAP
- Linux als Applikationsplatform/Runtime
  - siehe QNAP aber auch SONiC
- geschlossene Treiber bzw. IP[4]
- oftmals “Linux hostet das Frontend zum Switch-Chip”
- i.d.R. keinen Zugriff auf das (GNU/)Linux darunter
  - oftmals nichtmal GNU

[4] Intellectual Property

Docker/OCI

- Beispiele: MikroTik RouterOS, Nvidia ONYX
- Edge Computing *Lite*
- programmatische Interaktion mit Fabric
- Augmentation bestehender Features
- Monitoring (ein bisschen)
- *tcpdump* (ein bisschen)

DPDK

- DPDK: FUSE für Netzwerkhardware
- auch: Nvidia DOCA
  - like DPDK: API/Framework
  - unlike DPDK: Endergebnis läuft in Hardware
- Nutzer: FD.IO VPP
- kein Kernel send/notify/locking (DPDK $\Rightarrow$ Spinlock)
  - immer 1 CPU Kern voll ausgelastet
- dynamisch, alles Software
- Performancegrenzen
  - 16$\times$PCIe 4.0: $\sim$250 Gbit/s (nur DPDK)
  - CPU***s***

ONIE

- Beispiele: Supermicro, Nvidia
- “Fancy Grub & Rescue Shell”
- De- und Installieren von OSes
- genormte Umgebung
- User entscheiden über das OS
- ursprünglich von Cumulus Networks
- dafür auch der erwähnte USB Slot

Offene Switch Distributionen

- vollständige GNU/Linux Installation
- root Zugriff
- Möglichkeit von Modifikation
- Automatisierung <3
- Monitoring ~meh
- existierende Expertise nutzen
  - SysAdmins, die den Switch verstehen
  - Switch-Admins, die wissen wie der Server konfiguriert werden muss

Offene Switch Distributionen – Cumulus

- Vendor: Nvidia
- proprietär?
  - Cumulus, Mellanox&Broadcom, Nvidia
- Debian
- Configurationmanagement: NVUE – Nvidia User Experience Daemon
  - zentrale Konfiguration aller Netzwerkdienste
  - API (REST, gRPC?)
- *switchd*: SDK Magie
  - Verlagerung der Config in Hardware
- alles ohne Hardwareunterstützung läuft als GNU/Linux Daemon oder im Linux Kernel

Offene Switch Distributionen – SONiC

- Vendor: Microsoft
- Linux als Applikationsplatform/Runtime
- quasi alles in Docker-hosted Microservices
- Configurationmanagement: Redis
  - schneller Konfigurationswechsel
- Hardwareinterface via SAI – Switch Abstraction Interface
  - Kommerzieller Support
- “Vendor Lock-In” für OS

Offene Switch Distributionen – Dent

- Vendor: Amazon
- kurze Featureliste[5]
- effektiv: Hardwareoffloading seitens Linux
- $\Rightarrow$ siehe *switchdev*

[5] https://dent.dev/dentos/

Native Linux Treiber

- *switchdev*!
- L2 und L3 Offloading
- Treiber müssen es unterstützen: Nvidia & Marvell
- Bridge-Infrastruktur von Linux
- beliebige Distro, aktueller Kernel
  - Marvell sehr vorbildlich[6]
- Configurationmanagement: *whatever*
  - manuell, Ansible, Puppet, Salt, Nix, ...
- USB Slot!

[6] github.com/Marvell-switching/switchdev-prestera

Outro

Was bringt die Zukunft?

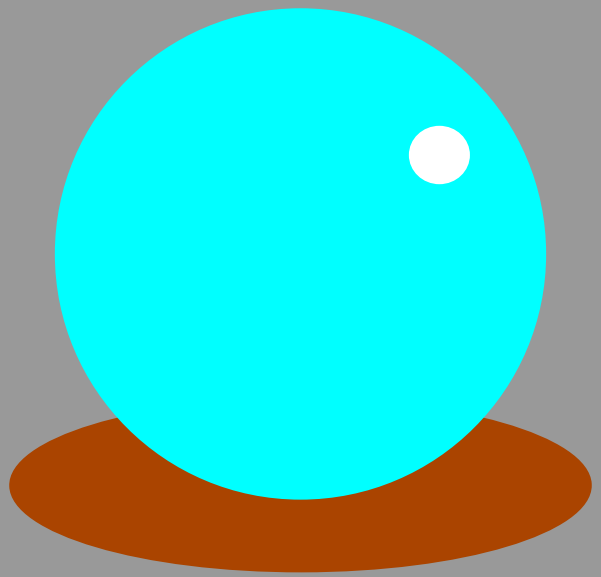

Abbildung 8: Creative Commons CC0 1.0 Universal Public Domain Dedication

Was bringt die Zukunft? – Cloud

- Zunahme von heterogener Infrastruktur & Mandantenfähigkeit
  - TCO bei Betreuung zweier unterschiedlicher Systeme recht hoch
- SDN jetzt schon integraler Bestandteil von Cloud
- Shift auf Routing statt Switching kommt auch langsam im HPC an
  - BUM Traffic gering halten
  - Migration von Workloads zur Laufzeit ermöglichen

Was bringt die Zukunft? – Hardware

- Flashspeicher günstiger: Switches mit mehr Disk
  - vollständiges OS
- schneller
  - 400G: aktuell im Einsatz
  - 800G: Nvidia Spectrum-4
- “besser”
  - mehr Fähigkeiten in Hardware (EVPN/OVS/IPv6/...)
- DPUs (Beispiel: Nvidia Bluefield)
  - für IaaS o.Ä.

Was bringt die Zukunft? – Software

- Softwarestack kommt bei 10G schon nicht mit
  - 400G, 800G $\Rightarrow$ mehr Hardwareintegration notwendig
- Hardwareintegration mindestens seitens Marvell und Nvidia
- Prosumer Bereich: MikroTik verbaut Marvell Chips
- EVPN und andere Erweiterungen von bestehenden Protokollen
- ~~mTLS~~

Was bringt die Zukunft? – Enterprise

- große Hersteller für Verlässlichkeit
- kommerzieller Support
- Cisco verliert Land
- Skalierbarkeit
  - Clos/Spine-Leaf Architekturen: Redundanz und ECMP

Tooling

- *pandoc*, LaTeX, Beamer
- *pdfpc*

Nochmal Ich

- privat
  - benary.org (IPv6 only)
  - @benaryorg@catcatnya.com
  - binary@benary.org
- beruflich
  - katze@menzel-it.net
  - menzel-it.net
- CLT Präsentation: clt2023.benary.org