

Programmierbare Netzwerke

Paradigmen, Tools und Einsatzgebiete

Prof. Dr. Zdravko Bozakov
bozakov@hs-worms.de

Forschung meets Hub, 31. März 2021



Intermediate Systems: Router und Switches bilden das Rückgrat des Internets.



Intermediate Systems: Router und Switches bilden das Rückgrat des Internets.



Hauptaufgabe

Weiterleitung von Paketen vom einem Eingangsport zu einem Ausgangsport

Paketvermittlung Internetverkehr wird in einzelnen Paketen übertragen

Paketvermittlung Internetverkehr wird in einzelnen Paketen übertragen

Routing (verteilter) Algorithmus, der die kürzesten Pfade zu jedem Ziel im Netzwerk berechnet

Paketvermittlung Internetverkehr wird in einzelnen Paketen übertragen

Routing (verteilter) Algorithmus, der die kürzesten Pfade zu jedem Ziel im Netzwerk berechnet

Forwarding Prozess der Pakete zum richtigen Ausgangsport weiterleitet

Paketvermittlung Internetverkehr wird in einzelnen Paketen übertragen

Routing (verteilter) Algorithmus, der die kürzesten Pfade zu jedem Ziel im Netzwerk berechnet

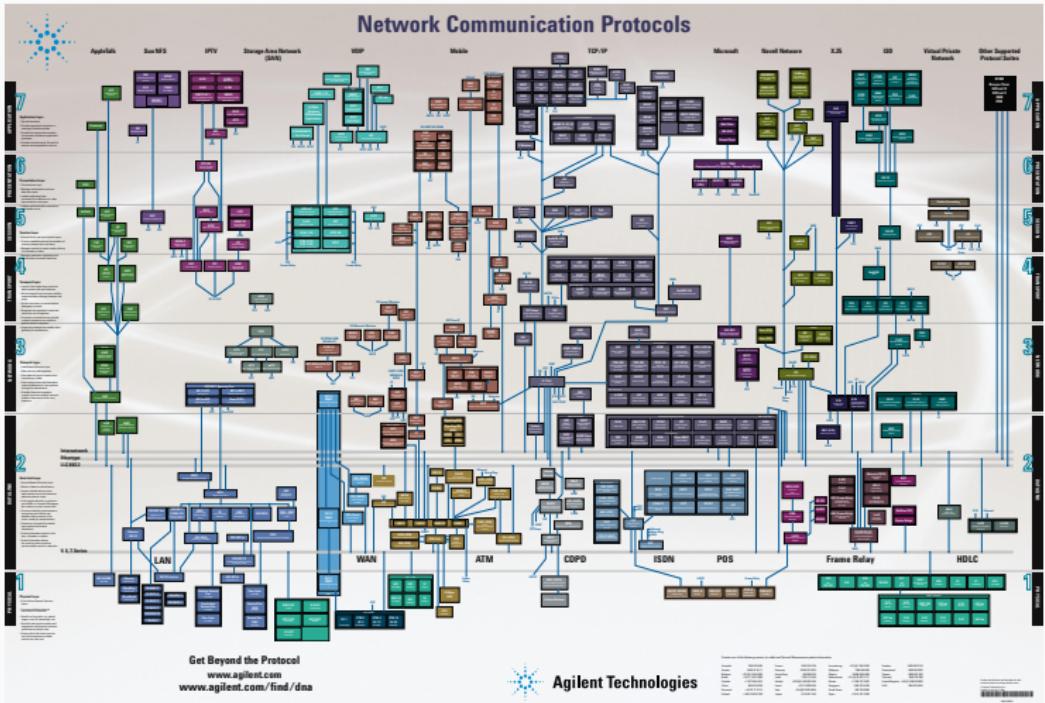
Forwarding Prozess der Pakete zum richtigen Ausgangsport weiterleitet

Queueing Mechanismus/Strategie um Pakete zu puffern um Überlasten abzufedern

Traditionell werden solche Geräte von wenigen spezialisierten Herstellern (Cisco, Juniper, Arista, Nokia, ...) entwickelt und (teuer) verkauft

Traditionell werden solche Geräte von wenigen spezialisierten Herstellern (Cisco, Juniper, Arista, Nokia, ...) entwickelt und (teuer) verkauft

Warum Eigentlich?



- Stabilität und Redundanz (Carrier-Grade)

- Stabilität und Redundanz (Carrier-Grade)
- Hohe Entwicklungskosten: diverse Protokolle, Spezialhardware, spezialisierte Betriebssysteme

- Stabilität und Redundanz (Carrier-Grade)
- Hohe Entwicklungskosten: diverse Protokolle, Spezialhardware, spezialisierte Betriebssysteme
- Hohe Geschwindigkeiten: bei 10 Gbps muss jedes Paket innerhalb 51 ns verarbeitet sein

- Stabilität und Redundanz (Carrier-Grade)
- Hohe Entwicklungskosten: diverse Protokolle, Spezialhardware, spezialisierte Betriebssysteme
- Hohe Geschwindigkeiten: bei 10 Gbps muss jedes Paket innerhalb 51 ns verarbeitet sein
- Hohe Port-Dichte: z.B.: 48 Ports \times 10 Gbps müssen im schlimmsten Fall 960 Gbps verarbeitet werden (oder \sim 2 Milliarden Pakete pro Sekunde)

Ist doch prima!?

Ist doch prima!?

Könnte besser sein ...

- “One-size-fits-all” Ansatz

Ist doch prima!?

Könnte besser sein ...

- “One-size-fits-all” Ansatz
- Monolithische Architektur

Ist doch prima!?

Könnte besser sein ...

- “One-size-fits-all” Ansatz
- Monolithische Architektur
- Stagnierende Innovation (Ossification)

Ist doch prima!?

Könnte besser sein ...

- “One-size-fits-all” Ansatz
- Monolithische Architektur
- Stagnierende Innovation (Ossification)
- Vendor-Lock-in

Ist doch prima!?

Könnte besser sein ...

- “One-size-fits-all” Ansatz
- Monolithische Architektur
- Stagnierende Innovation (Ossification)
- Vendor-Lock-in
- Komplexe Administration (CLI vs. DevOps)

Das Software-Defined-Networking Paradigma hat das Ziel die Frage zu beantworten:

Wie kann der Betrieb von Computernetzwerken verbessert werden und Innovationen vorangetrieben werden.

Das Software-Defined-Networking Paradigma hat das Ziel die Frage zu beantworten:

Wie kann der Betrieb von Computernetzwerken verbessert werden und Innovationen vorangetrieben werden.

...um neue Dienste effizient zu entwickeln und schnell in Betrieb zu nehmen

Grundidee

Erfolgsgeschichte des PCs nachahmen



[Wikimedia CC-BY-SA-3.0]

Grundidee

Erfolgsgeschichte des PCs nachahmen



[Wikimedia CC-BY-SA-3.0]

- Offene Software → freie Wahl des Betriebssystems und der Applikationen

Grundidee

Erfolgsgeschichte des PCs nachahmen



[Wikimedia CC-BY-SA-3.0]

- Offene Software → freie Wahl des Betriebssystems und der Applikationen
- Austauschbare Hardware → “whitebox” Switches

Abstraktionen sind in den Betriebssystem- und Software-Engineering Bereichen extrem erfolgreich ...

```

MONITOR FOR 6802 1.4      9-14-80 TSC ASSEMBLER PAGE  2

C000          ORG  ROM:$000000 BEGIN MONITOR
C000 8E 00 7D  START  LDS  #$STACK

*****  

* FUNCTION: INITA - Initialize ACIA  

* INPUT: none  

* OUTPUT: none  

* CALLS: none  

* DESTROYS: acc A  

*****  

0013    RESETA EQU  $00010011
0011    CLRACR EQU  $00010001

C005 87 80 04  INITA  LDA A #RESETA  RESET ACIA
C005 87 80 04        STA A ACIA
C004 86 11        LDA A #CLRACR  SET 8 BITS AND 2 STOP
C004 87 80 04        STA A ACIA

C00D 7E CD F1  JMC  SIGNON  GO TO START OF MONITOR
*****  

* FUNCTION: INCH - Input character
* INPUT: none
* OUTPUT: char in acc A
* DESTROYS: acc A
* CALLS: none
* DESTROYS: acc A
*****  

DESCRIPTION: Gets 1 character from terminal

C010 86 80 04  INCH  LDA A ACIA   GET STATUS
C011 47        ADD A #0      SHIFT RIPE FLAG INTO CARRY
C014 24 5A      RLC  INCH   RECEIVE NOT READY
C016 86 80 05  LDA A #0000+1  GET CHAR
C016 87 80 05        STA A RTSP  MOVE TO PORTIBILITY
C013 84 5F      JMP  OUTCH  ECHO & RTSP
C018 7E CD 79

*****  

* FUNCTION: INHEX -- INPUT HEX DIGIT
* INPUT: none
* OUTPUT: digit in acc A
* CALLS: none
* DESTROYS: acc A
* Returns to monitor if not HEX input
*****  

C01E 8D F0  INHEX  RRN  INCH   GET A CHAR
C01F 81 3D        CMP A #0      ZERO
C020 81 12        CMP A #FF    NOT HEX
C024 81 39        CMP A #'9     NINE
C024 2F 0A        HLT  NHEXITS  GOOD HEX
C025 81 31        CMP A #'A     BADHEX
C024 2B 09        RNC  HEXERR  NOT HEX
C02C 81 46        CMP A #'F     BADHEX
C02D 81 40        RNC  HEXERR  NOT HEX
C030 80 07        SUB A #7    FIX A-F
C032 84 0F        HERRITS  AND A #$0F  CONVERT ASCII TO DIGIT
C034 39        RTS

C035 7E CO AF  HERRERR  JMF  CTRL  RETURN TO CONTROL LOOP

```

[Wikimedia]

- Komplexität wird durch Abstraktionen verborgen (z.B. virtueller Speicher, Dateisysteme, virtuelle Maschinen)
- Ermöglichen Automatisierung/Optimierung (Compiler, Caching, Multitasking)
- Passende Tools für unterschiedliche Probleme (Assembler vs. C vs. Fortran vs. Java vs. Python)
- Entwickler/User können sich auf die Lösung von Problemen konzentrieren
- Wesentlich schnellerer Entwicklungszyklus im Softwarebereich

Der SDN Ansatz wurde maßgeblich an US Universitäten Standford und Princeton entwickelt^[1]

- Forschung als Haupttreiber

^[1] McKeown et al. "OpenFlow: Enabling Innovation in Campus Networks", SIGCOMM-CCR, 2008

Der SDN Ansatz wurde maßgeblich an US Universitäten Standford und Princeton entwickelt^[1]

- Forschung als Haupttreiber
- Lobbying der Hardwarehersteller (Secret Sauce)

^[1] McKeown et al. "OpenFlow: Enabling Innovation in Campus Networks", SIGCOMM-CCR, 2008

Der SDN Ansatz wurde maßgeblich an US Universitäten Standford und Princeton entwickelt^[1]

- Forschung als Haupttreiber
- Lobbying der Hardwarehersteller (Secret Sauce)
- Realistische Architektur

^[1] McKeown et al. "OpenFlow: Enabling Innovation in Campus Networks", SIGCOMM-CCR, 2008

Das SDN Modell führt drei wesentliche Abstraktionen ein:

Das SDN Modell führt drei wesentliche Abstraktionen ein:

Forwarding Implementierungsdetails des Datenpfades sind vom Kontrollprogramm verborgen

Das SDN Modell führt drei wesentliche Abstraktionen ein:

Forwarding Implementierungsdetails des Datenpfades sind vom Kontrollprogramm verborgen

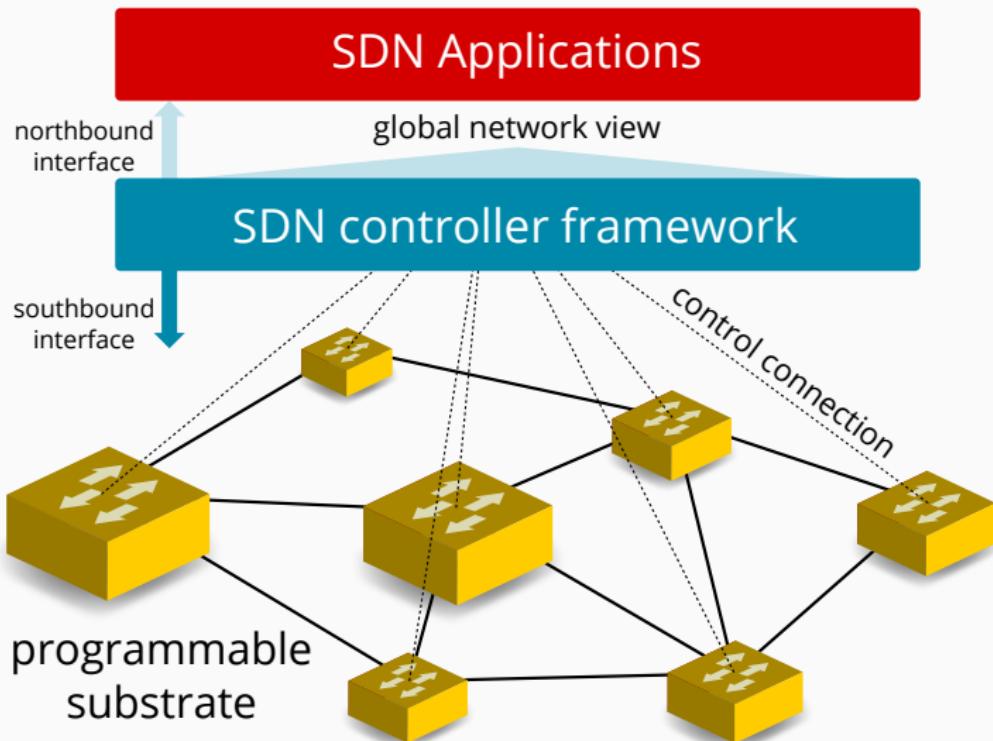
Globale Netzwerksicht Allen Diensten steht eine globale Sicht des Netzwerks zur Verfügung

Das SDN Modell führt drei wesentliche Abstraktionen ein:

Forwarding Implementierungsdetails des Datenpfades sind vom Kontrollprogramm verborgen

Globale Netzwerksicht Allen Diensten steht eine globale Sicht des Netzwerks zur Verfügung

Spezifikation Das Verhalten von Netzwerkdiensten wird durch ein (zentralisiertes) Kontrollprogramm definiert



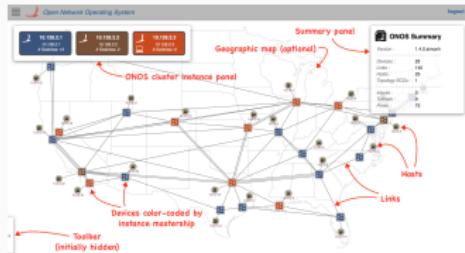
Was kann man damit machen?

- Höhere Programmiersprachen zur Beschreibung von Netzwerklogik und Konnektivität (Compiler/Debugger z.B. Frenetic)
- Automatisierte Verifikation von Netzwerkregeln (z.B. ACL)
- Transparente Migration virtueller Maschinen

Interoperabilität und offene Schnittstellen sind wesentlicher Bestandteil der SDN Idee

Es existieren mehrere Open Source Controller mit unterschiedlichen Schwerpunkten

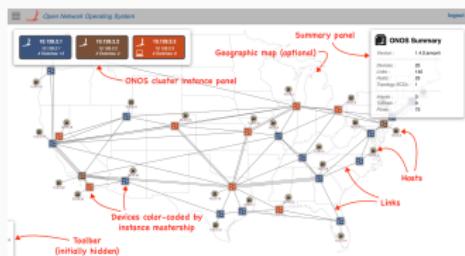
- ONOS (Java)
- OpenDaylight (Java)
- Faucet, Ryu (Python)
- Trema (Ruby)
- u.v.m. ...



Interoperabilität und offene Schnittstellen sind wesentlicher Bestandteil der SDN Idee

Es existieren mehrere Open Source Controller mit unterschiedlichen Schwerpunkten

- ONOS (Java)
- OpenDaylight (Java)
- Faucet, Ryu (Python)
- Trema (Ruby)
- u.v.m. ...



...sowie kommerzielle "SDN" Produkte

- VMware NSX-T, Cisco APIC, Tungsten Fabric, Big Switch, Pluribus, ...

Zudem haben sich eine Reihe offener und kommerzieller Switch-Betriebssysteme etabliert

- Open Network Linux (Open Compute Project)
- SONiC (Microsoft Azure)
- baseboxd
- Cumulus Linux (Nvidia)
- NoviFlow
- PICOS
- ...



[Quelle: Facebook]

- White-Box Switches:^[2] Edge-Core, Dell, Mellanox, DNI, Stordis, ...
- SmartNICs (Intel, Netronome, Nvidia)
- Open Compute Project (Wedge, Open GPON-OLT, ...)



[Quelle: Intel]

[2] <https://opennetlinux.org/hcl.html>

OpenFlow

Was ist OpenFlow?

Offene Schnittstelle zur externen Programmierung
der Forwardingtabellen in kommerziellen Switches

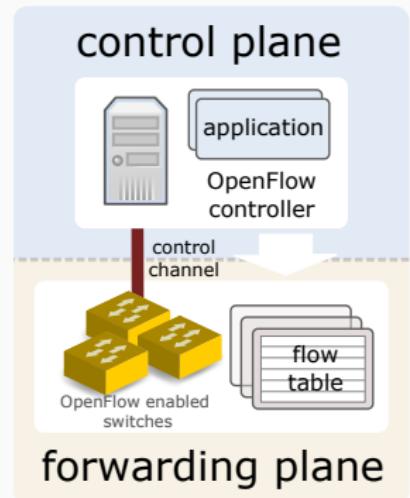
Was ist OpenFlow?

Offene Schnittstelle zur externen Programmierung
der Forwardingtabellen in kommerziellen Switches

(Southbound-Interface für SDN)

OpenFlow lagert die Kontrolllogik von kommerziellen Switches auf einen externen Controller aus

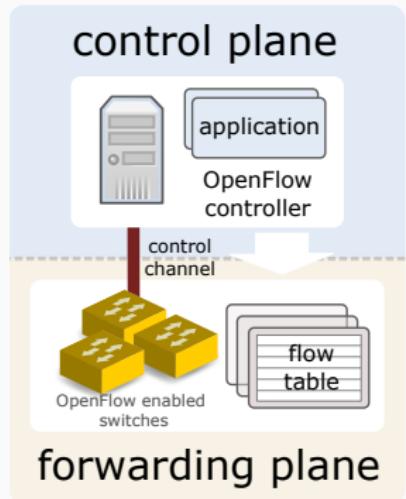
- Einfache Geräte mit programmierbaren Forwardingtabellen
- Logisch zentralisierter Controller
- Kommunikation über Kontrollkanal (SSL, in-band/out-of-band)



[3] <https://opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf>

OpenFlow lagert die Kontrolllogik von kommerziellen Switches auf einen externen Controller aus

- Einfache Geräte mit programmierbaren Forwardingtabellen
- Logisch zentralisierter Controller
- Kommunikation über Kontrollkanal (SSL, in-band/out-of-band)



OpenFlow spezifiziert eine API sowie das Kommunikationprotokoll^[3] und wird von der Open Networking Foundation verwaltet

[3] <https://opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf>

Aus Sicht des Controllers ist jeder OpenFlow Switch eine Flow-Tabelle

Jede Tabelle wird mit "Flow-Einträge" gefüllt:

Match Header-Bitfolge mit der ankommende Pakete verglichen werden

OpenFlow header fields								
input port	Ethernet source	Ethernet dest.	Ethernet type	VLAN ID	IP source	IP dest.	IP protocol	source port

Aus Sicht des Controllers ist jeder OpenFlow Switch eine Flow-Tabelle

Jede Tabelle wird mit "Flow-Einträge" gefüllt:

Match Header-Bitfolge mit der ankommende Pakete verglichen werden

OpenFlow header fields									
input port	Ethernet source	Ethernet dest.	Ethernet type	VLAN ID	IP source	IP dest.	IP protocol	source port	dest. port

Actions Forwarding- und Verarbeitungs-Operationen, die auf jedes passende Paket ausgeführt werden sollen
z.B.: drop, out_port, send_to_controller, modify_IP_dst, set_VLAN_ID, ...

Aus Sicht des Controllers ist jeder OpenFlow Switch eine Flow-Tabelle

Jede Tabelle wird mit "Flow-Einträge" gefüllt:

Match Header-Bitfolge mit der ankommende Pakete verglichen werden

OpenFlow header fields									
input port	Ethernet source	Ethernet dest.	Ethernet type	VLAN ID	IP source	IP dest.	IP protocol	source port	dest. port

Actions Forwarding- und Verarbeitungs-Operationen, die auf jedes passende Paket ausgeführt werden sollen

z.B.: drop, out_port, send_to_controller,
modify_IP_dst, set_VLAN_ID, ...

Counters zählen das Datenvolumen des "Flows"

OpenFlow ist zwar sehr flexibel, hat jedoch einige Beschränkungen:

- Flow-Tabellen verwenden vordefinierte Headerfelder (Versionsabhängig)
- OpenFlow Aktionen sind vorgegeben (drop, forwards, rewrite, ...)

[3] Bosshart et al. "P4: Programming Protocol-Independent Packet Processors", ACM CCR, 2014

OpenFlow ist zwar sehr flexibel, hat jedoch einige Beschränkungen:

- Flow-Tabellen verwenden vordefinierte Headerfelder (Versionsabhängig)
- OpenFlow Aktionen sind vorgegeben (drop, forwards, rewrite, ...)



P4 ist eine Sprache für die Programmierung des Datenpfads:

[3] Bosshart et al. "P4: Programming Protocol-Independent Packet Processors", ACM CCR, 2014

OpenFlow ist zwar sehr flexibel, hat jedoch einige Beschränkungen:

- Flow-Tabellen verwenden vordefinierte Headerfelder (Versionsabhängig)
- OpenFlow Aktionen sind vorgegeben (drop, forwards, rewrite, ...)



P4 ist eine Sprache für die Programmierung des Datenpfads:

- Dynamische Rekonfiguration

[3] Bosshart et al. "P4: Programming Protocol-Independent Packet Processors", ACM CCR, 2014

OpenFlow ist zwar sehr flexibel, hat jedoch einige Beschränkungen:

- Flow-Tabellen verwenden vordefinierte Headerfelder (Versionsabhängig)
- OpenFlow Aktionen sind vorgegeben (drop, forwards, rewrite, ...)



P4 ist eine Sprache für die Programmierung des Datenpfads:

- Dynamische Rekonfiguration
- Protokollunabhängig (Paketheader & eigene Aktionen)

[3] Bosshart et al. "P4: Programming Protocol-Independent Packet Processors", ACM CCR, 2014

OpenFlow ist zwar sehr flexibel, hat jedoch einige Beschränkungen:

- Flow-Tabellen verwenden vordefinierte Headerfelder (Versionsabhängig)
- OpenFlow Aktionen sind vorgegeben (drop, forwards, rewrite, ...)



P4 ist eine Sprache für die Programmierung des Datenpfads:

- Dynamische Rekonfiguration
- Protokollunabhängig (Paketheader & eigene Aktionen)
- Hardwareunabhängig (Kommerzielle Switches, Software-Switches, SmartNICs)

[3] Bosshart et al. "P4: Programming Protocol-Independent Packet Processors", ACM CCR, 2014

Experimentieren mit SDN

Wie kann man SDN ausprobieren?

Wie kann man SDN ausprobieren?

- ➊ Open vSwitch (Linux Bridge mit OpenFlow-Support)

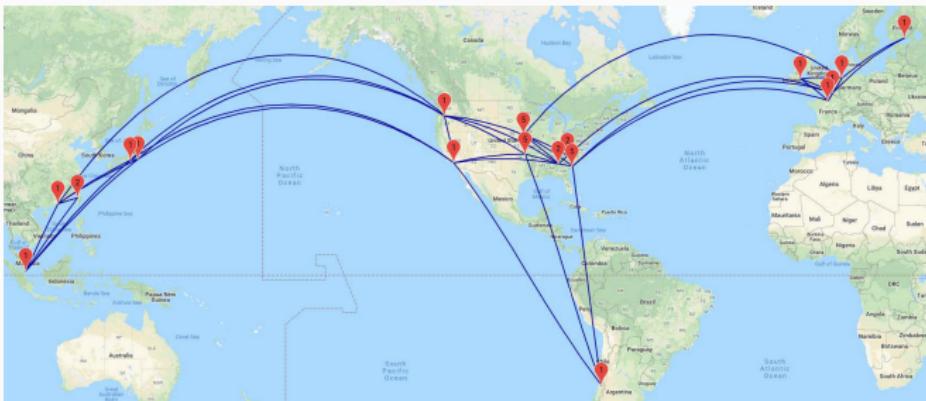
Wie kann man SDN ausprobieren?

- ① **Open vSwitch** (Linux Bridge mit OpenFlow-Support)
- ② **Mininet** (Netzwerk Emulator)

Wie kann man SDN ausprobieren?

- ① **Open vSwitch** (Linux Bridge mit OpenFlow-Support)
- ② **Mininet** (Netzwerk Emulator)
- ③ **SDN Controller** (z.B. ONOS)

SDN Projekte & Use-Cases



- Google betreibt seit ca. 2011 ein WAN auf SDN Basis um die eigenen Rechenzentren zu verbinden^{[4][5]}

[4] B4: Experience with a Globally Deployed Software Defined WAN

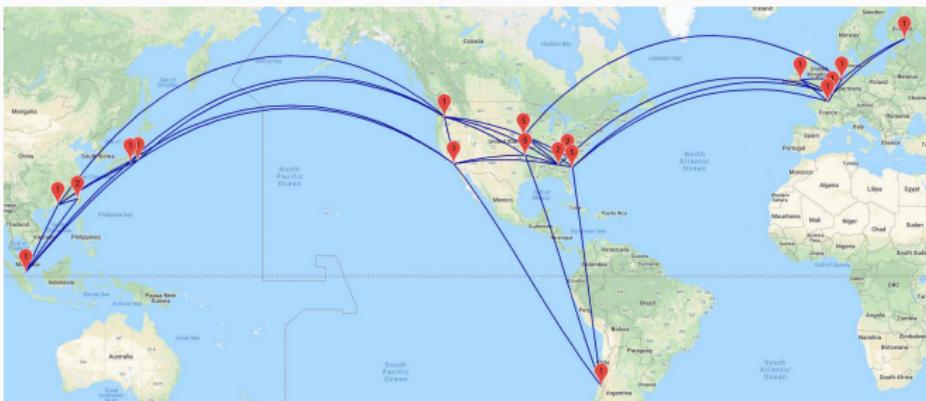
[5] B4 and After: Managing Hierarchy, Partitioning, and Asymmetry for Availability and Scale in Google's Software-Defined WAN



- Google betreibt seit ca. 2011 ein WAN auf SDN Basis um die eigenen Rechenzentren zu verbinden^{[4][5]}
- 33 Sites, verschiedenen Verkehrsklassen, fast 100% Link-Auslastung, Verfügbarkeit von 99.99%, Verdopplung der Bandbreite alle 9 Monate

[4] B4: Experience with a Globally Deployed Software Defined WAN

[5] B4 and After: Managing Hierarchy, Partitioning, and Asymmetry for Availability and Scale in Google's Software-Defined WAN



- Google betreibt seit ca. 2011 ein WAN auf SDN Basis um die eigenen Rechenzentren zu verbinden^{[4][5]}
- 33 Sites, verschieden Verkehrsklassen, fast 100% Link-Auslastung, Verfügbarkeit von 99.99%, Verdopplung der Bandbreite alle 9 Monate
- Selbstgebaute Switches, Traffic Engineering Algorithmen und Softwarekomponenten

[4] B4: Experience with a Globally Deployed Software Defined WAN

[5] B4 and After: Managing Hierarchy, Partitioning, and Asymmetry for Availability and Scale in Google's Software-Defined WAN

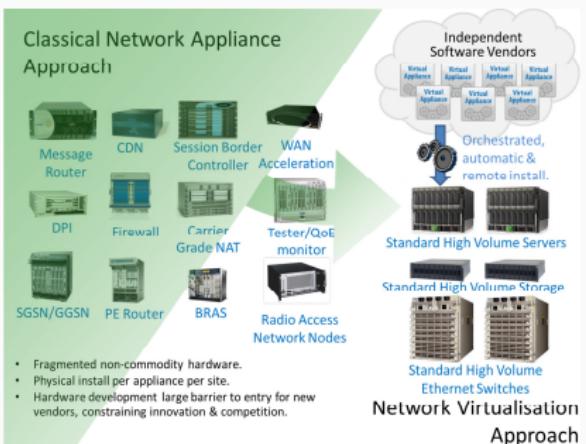
Die Vorteile von SDN wurden auch von der Telekommunikationsbranche erkannt

[6] Network Functions Virtualisation – Introductory White Paper, 2012

[7] Network Functions Virtualisation – Update White Paper, 2013

Die Vorteile von SDN wurden auch von der Telekommunikationsbranche erkannt

- In 2012 wurde auf dem "SDN and OpenFlow World Congress" in Darmstadt das NFV Konzept vorgestellt [6] [7]

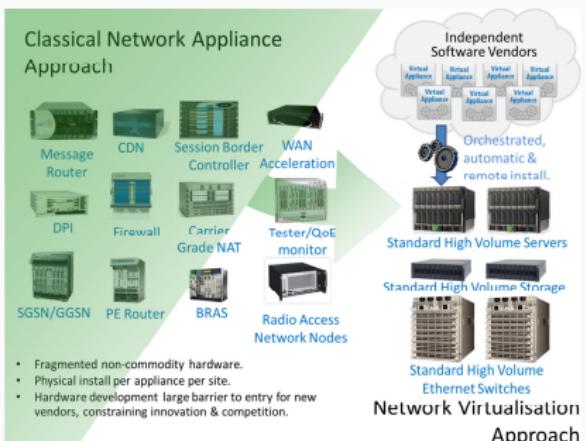


[6] Network Functions Virtualisation – Introductory White Paper, 2012

[7] Network Functions Virtualisation – Update White Paper, 2013

Die Vorteile von SDN wurden auch von der Telekommunikationsbranche erkannt

- In 2012 wurde auf dem "SDN and OpenFlow World Congress" in Darmstadt das NFV Konzept vorgestellt [6] [7]
- Kurz darauf wurde die ETSI ISG NFV gegründet
(<https://www.etsi.org/committee/nfv>)



[6] Network Functions Virtualisation – Introductory White Paper, 2012

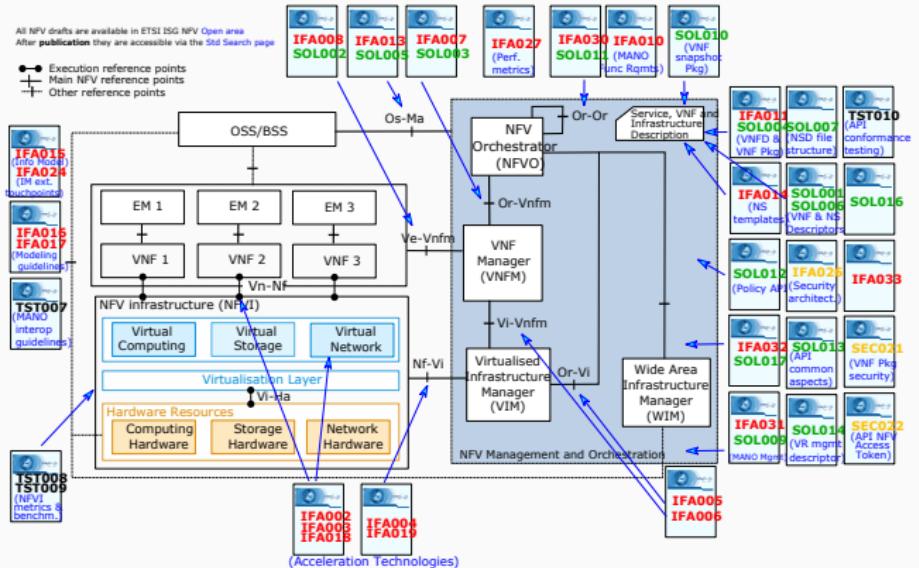
[7] Network Functions Virtualisation – Update White Paper, 2013

- Physikalische Netzwerkkomponenten durch virtuelle ersetzen (VNFs)

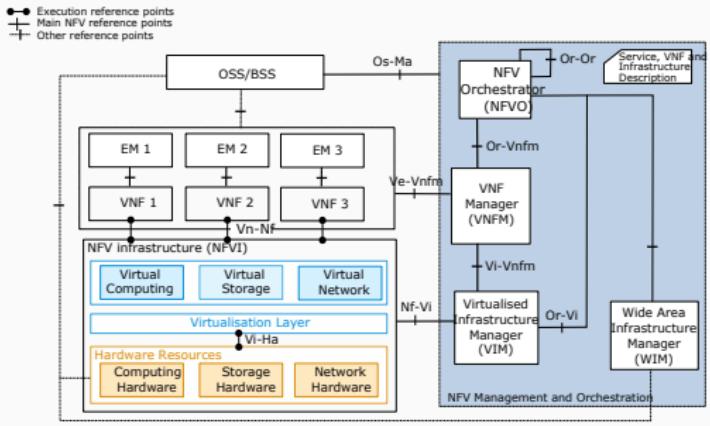
- Physikalische Netzwerkkomponenten durch virtuelle ersetzen (VNFs)
- Management und Lifecycle der Komponenten vereinfachen

- Physikalische Netzwerkkomponenten durch virtuelle ersetzen (VNFs)
- Management und Lifecycle der Komponenten vereinfachen
- Dienste und deren Abhangigkeiten modellieren

- Physikalische Netzwerkkomponenten durch virtuelle ersetzen (VNFs)
- Management und Lifecycle der Komponenten vereinfachen
- Dienste und deren Abhangigkeiten modellieren
- Interoperabilitat ermoglichen (NFVI, OSS)

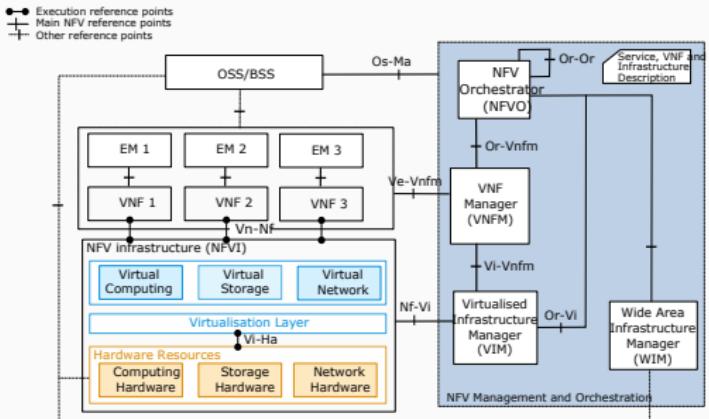


[8]



[8]

[8] <https://www.etsi.org/technologies/nfv>



[8]

Management and Orchestrierung (MANO):

- Virtualized Infrastructure Manager (VIM)
- VNF Manager
- NFV Orchestrator

[8] <https://www.etsi.org/technologies/nfv>

OSM^[9] ist eine quelloffene Implementierung von ETSI NFV MANO

Unterstützt unter anderem:

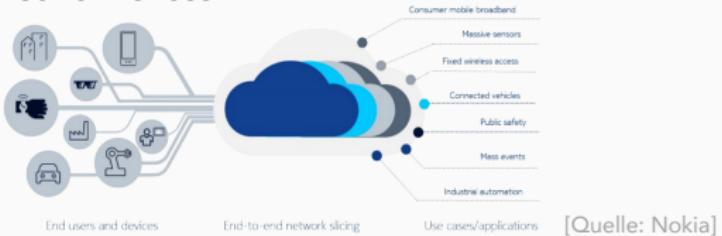
[9] <https://osm.etsi.org/>

[10] **SliceNet 5G PPP:** Wang, Bozakov, et al. "SliceNet: End-to-End Cognitive Network Slicing and Slice Management Framework in Virtualised Multi-Domain, Multi-Tenant 5G Networks", IEEE BMSB 2021

OSM^[9] ist eine quelloffene Implementierung von ETSI NFV MANO

Unterstützt unter anderem:

- Network Slices^[10]



[Quelle: Nokia]

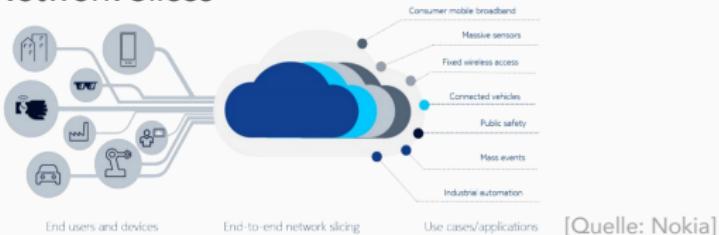
[9] <https://osm.etsi.org/>

[10] **SliceNet 5G PPP**: Wang, Bozakov, et al. "SliceNet: End-to-End Cognitive Network Slicing and Slice Management Framework in Virtualised Multi-Domain, Multi-Tenant 5G Networks", IEEE BMSB 2021

OSM^[9] ist eine quelloffene Implementierung von ETSI NFV MANO

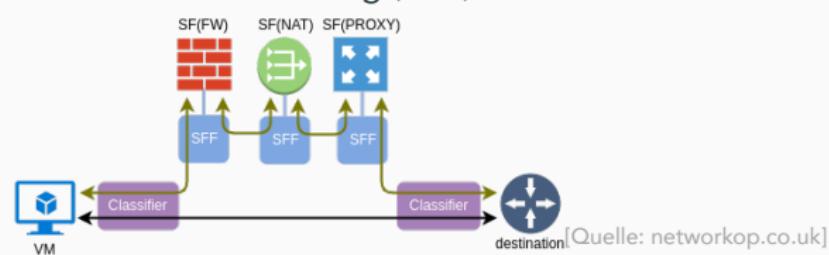
Unterstützt unter anderem:

- Network Slices^[10]



[Quelle: Nokia]

- Service Function Chaining (SFC)



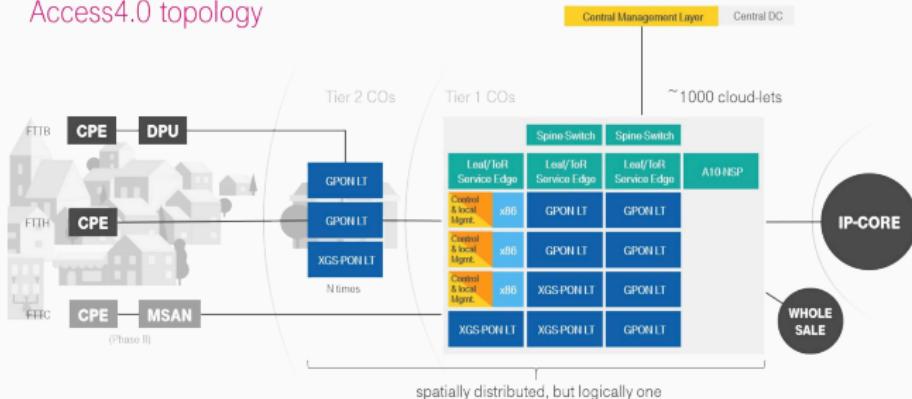
[Quelle: networkop.co.uk]

[9] <https://osm.etsi.org/>

[10] **SliceNet 5G PPP:** Wang, Bozakov, et al. "SliceNet: End-to-End Cognitive Network Slicing and Slice Management Framework in Virtualised Multi-Domain, Multi-Tenant 5G Networks", IEEE BMSB 2021

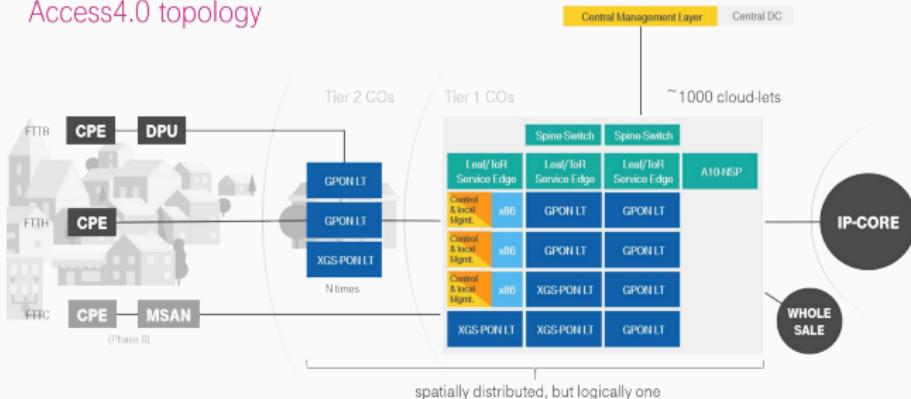
Software-Defined Access Network: "develop a cost-efficient, lean-to-operate and scalable access platform to deliver Gigabit products"

Access4.0 topology



Software-Defined Access Network: "develop a cost-efficient, lean-to-operate and scalable access platform to deliver Gigabit products"

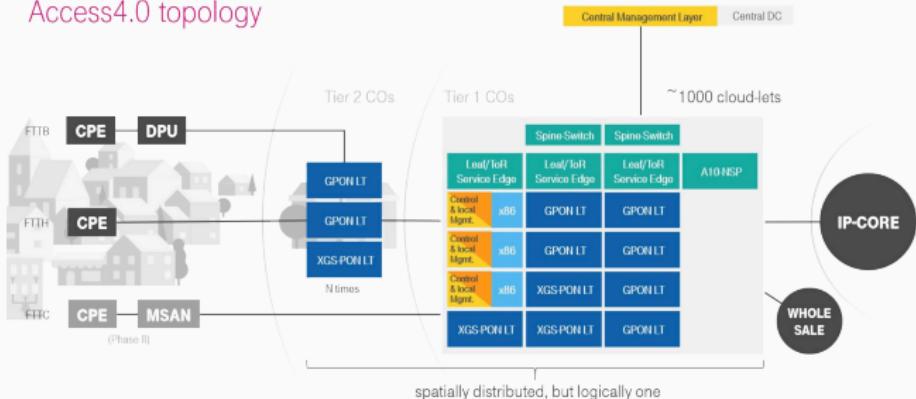
Access4.0 topology



- Vendor Lock eliminieren (offene Hardware & Software)

Software-Defined Access Network: "develop a cost-efficient, lean-to-operate and scalable access platform to deliver Gigabit products"

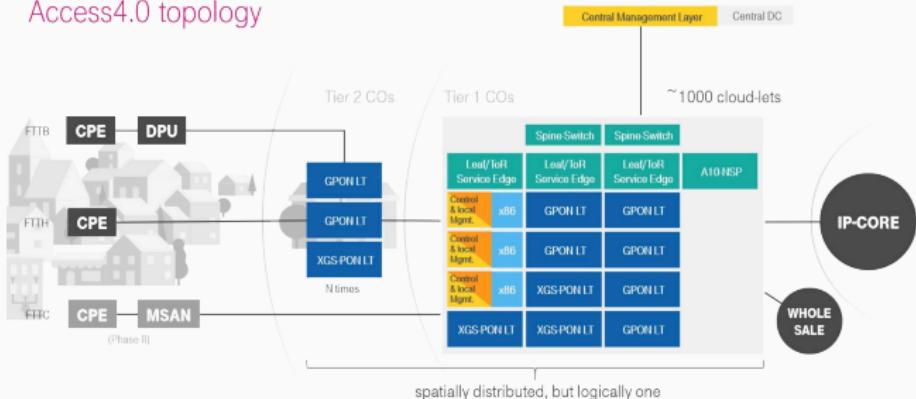
Access4.0 topology



- Vendor Lock eliminieren (offene Hardware & Software)
- Grad der Automatisierung erhöhen (weniger Personal, Testing)

Software-Defined Access Network: "develop a cost-efficient, lean-to-operate and scalable access platform to deliver Gigabit products"

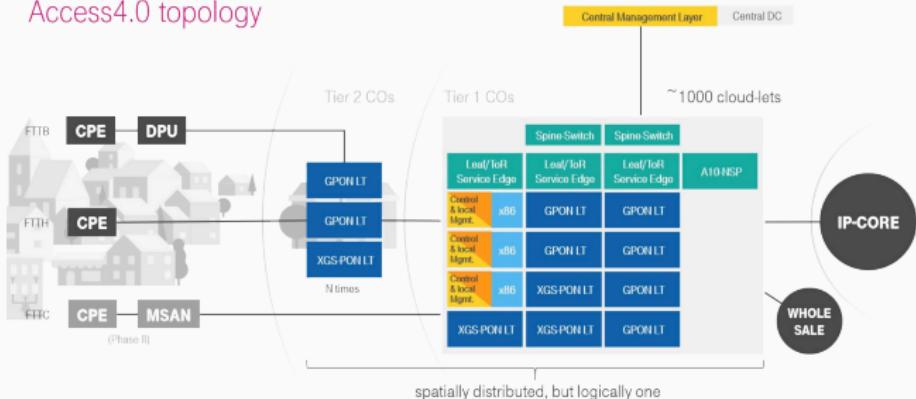
Access4.0 topology



- Vendor Lock eliminieren (offene Hardware & Software)
- Grad der Automatisierung erhöhen (weniger Personal, Testing)
- Time-to-Market reduzieren (Openness, CI/CD)

Software-Defined Access Network: "develop a cost-efficient, lean-to-operate and scalable access platform to deliver Gigabit products"

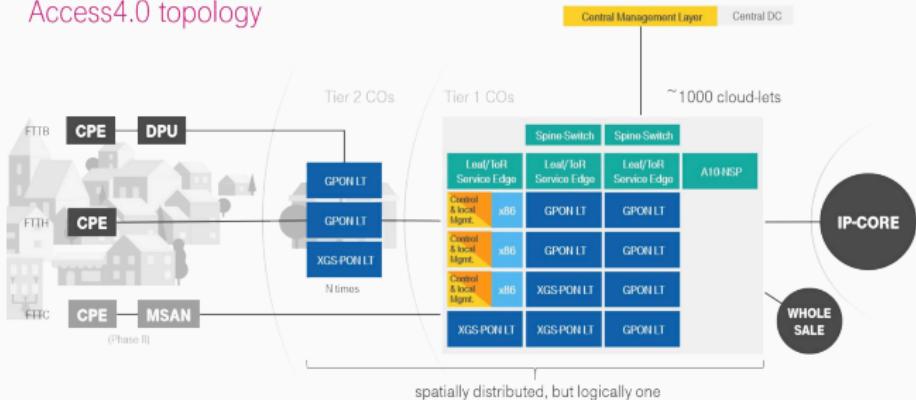
Access4.0 topology



- Vendor Lock eliminieren (offene Hardware & Software)
- Grad der Automatisierung erhöhen (weniger Personal, Testing)
- Time-to-Market reduzieren (Openness, CI/CD)
- Data-Center Prinzipien anwenden

Software-Defined Access Network: "develop a cost-efficient, lean-to-operate and scalable access platform to deliver Gigabit products"

Access4.0 topology



- Vendor Lock eliminieren (offene Hardware & Software)
- Grad der Automatisierung erhöhen (weniger Personal, Testing)
- Time-to-Market reduzieren (Openness, CI/CD)
- Data-Center Prinzipien anwenden

Kollaboration mit vielen Partnern (RtBrick, Reply, Broadcom, Adtran, Siticom, BISDN, Radisys, OneIT, Fibre Industries, ONF, ...)

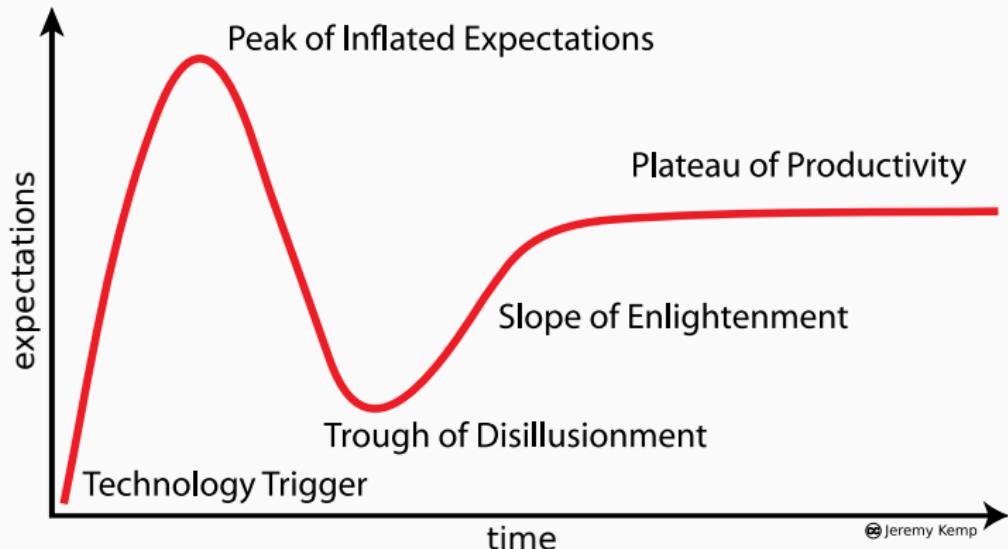
Herausforderungen und Ausblick

Natürlich existieren noch Herausforderungen und offene Forschungsfragen:

- Konsistenz der globalen Netzwerksicht muss garantiert werden
- Skalierbarkeit von (verteilten) SDN Controllern
- Kein einheitliches SDN Northbound-Interface

Natürlich existieren noch Herausforderungen und offene Forschungsfragen:

- Konsistenz der globalen Netzwerksicht muss garantiert werden
 - Skalierbarkeit von (verteilten) SDN Controllern
 - Kein einheitliches SDN Northbound-Interface
-
- Interoperabilität zwischen OpenFlow Herstellerimplementierungen
 - Konvergenz der Networking und Software-Engineering Communities
 - Vendor Lock-in → Product Lock-in, Skills Lock-in, etc.?



© Jeremy Kemp

- SDN hat einen Milliarden-Markt "geknackt"^[11]
- Open-source spielt eine zentrale Rolle
- Networking ist keine Insel: Synergien (u.A.) mit der Verteilte-Systeme, Software-Engineering, DevOps Communities
- Auch kleinere Player können von neuen Markt und Technologien profitieren
 - Datenverarbeitung ins Netzwerk "auslagern"
 - Speziallösungen für Telco-Indstrie

[11] <https://www.statista.com/statistics/468636/global-sdn-market-size/>

- Ziel: SDN Forschung und Lehre vorantreiben

[12] <https://www.hs-worms.de/csrg/>

- Ziel: SDN Forschung und Lehre vorantreiben
- Open Networking Lab@HSWO (Programmable Switches, SmartNICs)

[12] <https://www.hs-worms.de/csrg/>

- Ziel: SDN Forschung und Lehre vorantreiben
- Open Networking Lab@HSWO (Programmable Switches, SmartNICs)
- Cyber Security Research Group (CSRG)^[12]

[12] <https://www.hs-worms.de/csrg/>

- Ziel: SDN Forschung und Lehre vorantreiben
- Open Networking Lab@HSWO (Programmable Switches, SmartNICs)
- Cyber Security Research Group (CSRG)^[12]
- Erste Projekte:
 - Sicherung und Monitoring der Campusinfrastruktur
 - KISAT: "KI gestütztes Altern für eine Pandemie-resiliente digitale Infrastruktur"

[12] <https://www.hs-worms.de/csrg/>

- Ziel: SDN Forschung und Lehre vorantreiben
- Open Networking Lab@HSWO (Programmable Switches, SmartNICs)
- Cyber Security Research Group (CSRG)^[12]
- Erste Projekte:
 - Sicherung und Monitoring der Campusinfrastruktur
 - KISAT: "KI gestütztes Altern für eine Pandemie-resiliente digitale Infrastruktur"

[12] <https://www.hs-worms.de/csrg/>

- Ziel: SDN Forschung und Lehre vorantreiben
- Open Networking Lab@HSWO (Programmable Switches, SmartNICs)
- Cyber Security Research Group (CSRG)^[12]
- Erste Projekte:
 - Sicherung und Monitoring der Campusinfrastruktur
 - KISAT: "KI gestütztes Altern für eine Pandemie-resiliente digitale Infrastruktur"

Ich freue mich über Anregungen und Kooperationen!

Vielen Dank für Ihre Aufmerksamkeit!

[12] <https://www.hs-worms.de/csrg/>

Backup Slides

“Für allen IP Paketen mit TCP Port 8182 und dem Ziel 10.1.1.0/24, ändere die L2-Adressen und leite sie gleichzeitig an Ports 1 und 2 weiter!”

Datapath ID: 1234567890 ofp_flow_mod: OFPC_ADD									
OpenFlow header fields									
input port	Ethernet source	Ethernet dest.	Ethernet type	VLAN ID	IP source	IP dest.	IP protocol	source port	dest. port
*	*	*	0x0800	*	*	10.1.1.0/24	0x06	*	8182
OpenFlow actions									
modify eth src: 0a:76:06:ff:ff:00									
modify eth dst: 00:4e:46:32:43:00									
out port: 1									
out port: 3									

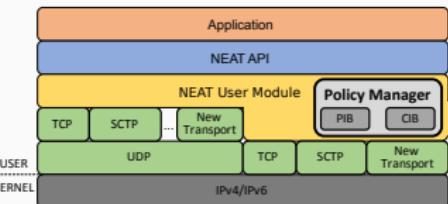
Controller generiert eine OF-Protokoll-Nachricht `ofp_flow_add` und sendet diese über den Kontrollkanal an den Switch

- Forschungsprojekt EU Horizon 2020
(**Dell EMC**, Mozilla, Cisco, Simula, KaU, FH-Münster)

[13] IETF TAPS working group

[14] Bozakov et al. "A NEAT framework for enhanced end-host integration in SDN environments", IEEE NFV-SDN 2017

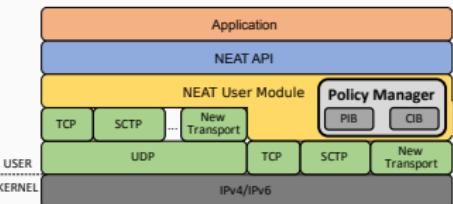
- Forschungsprojekt EU Horizon 2020
(Dell EMC, Mozilla, Cisco, Simula, KaU, FH-Münster)
- Ziel: moderne API für
Netzwerkapplikationen [13]



[13] IETF TAPS working group

[14] Bozakov et al. "A NEAT framework for enhanced end-host integration in SDN environments", IEEE NFV-SDN 2017

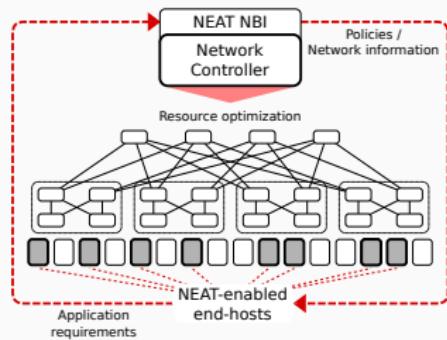
- Forschungsprojekt EU Horizon 2020
(Dell EMC, Mozilla, Cisco, Simula, KaU, FH-Münster)
- Ziel: moderne API für
Netzwerkapplikationen [13]
- Unterschiedliche Protokolle für
unterschiedliche Einsatzszenarien



[13] IETF TAPS working group

[14] Bozakov et al. "A NEAT framework for enhanced end-host integration in SDN environments", IEEE NFV-SDN 2017

- Forschungsprojekt EU Horizon 2020
(Dell EMC, Mozilla, Cisco, Simula, KaU, FH-Münster)
- Ziel: moderne API für Netzwerkapplikationen [13]
- Unterschiedliche Protokolle für unterschiedliche Einsatzszenarien
- Beispiel SDN Datacenter: Getrennte Behandlung von Bulk- und Low-latency-Flows. [14]

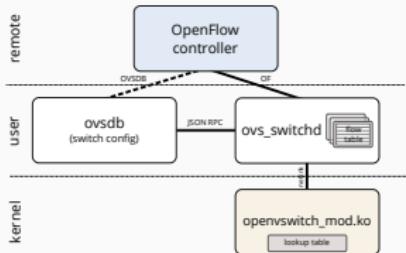


[13] IETF TAPS working group

[14] Bozakov et al. "A NEAT framework for enhanced end-host integration in SDN environments", IEEE NFV-SDN 2017

Open vSwitch^[15] (ovs) ist ein quelloffener Software-Switch

- “drop-in” Alternative zur Linux Bridge
- Unterstützt eine Reihe von Management- und Netzwerk-Technologien (NetFlow, sFlow, **OpenFlow**, SPAN, GRE, VXLAN, LACP, ...)
- Kernelmodul seit Version 3.3 im Mainline Linux Kernel



[15] <https://www.openvswitch.org/>

- Mininet^[16] erstellt virtuelle Topologien auf einer Maschine (Namespaces, cgroups)
- Virtuelle Hosts werden mit Open vSwitches verbunden
- Switches können mit beliebigen externen OpenFlow Controllern gesteuert werden
- Auf den virtuellen Hosts kann beliebiger Code ausgeführt werden (auch interaktiv)
- Scripting mit Python
- Anbindung an reales Netzwerk möglich

[16] mininet.org