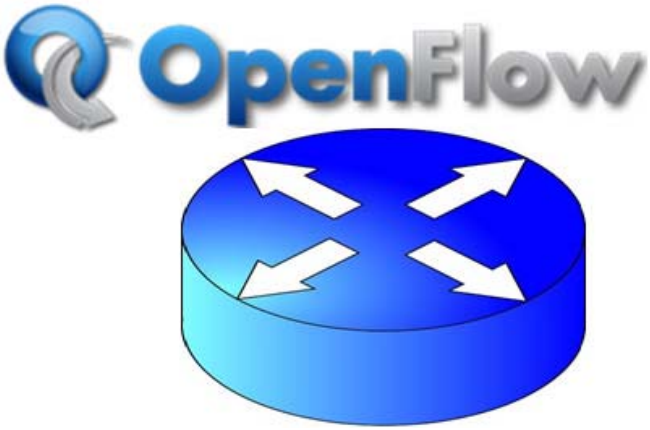
Software Defined Networking

Introduction

David Hausheer

Department of Electrical Engineering
and Information Technology
Technische Universität Darmstadt

E-Mail: hausheer@ps.tu-darmstadt.de
http://www.ps.tu-darmstadt.de/teaching/sdn



*Original slides for this lecture provided by Xenofontas Dimitropoulos (ETH Zurich)

Lecture Overview

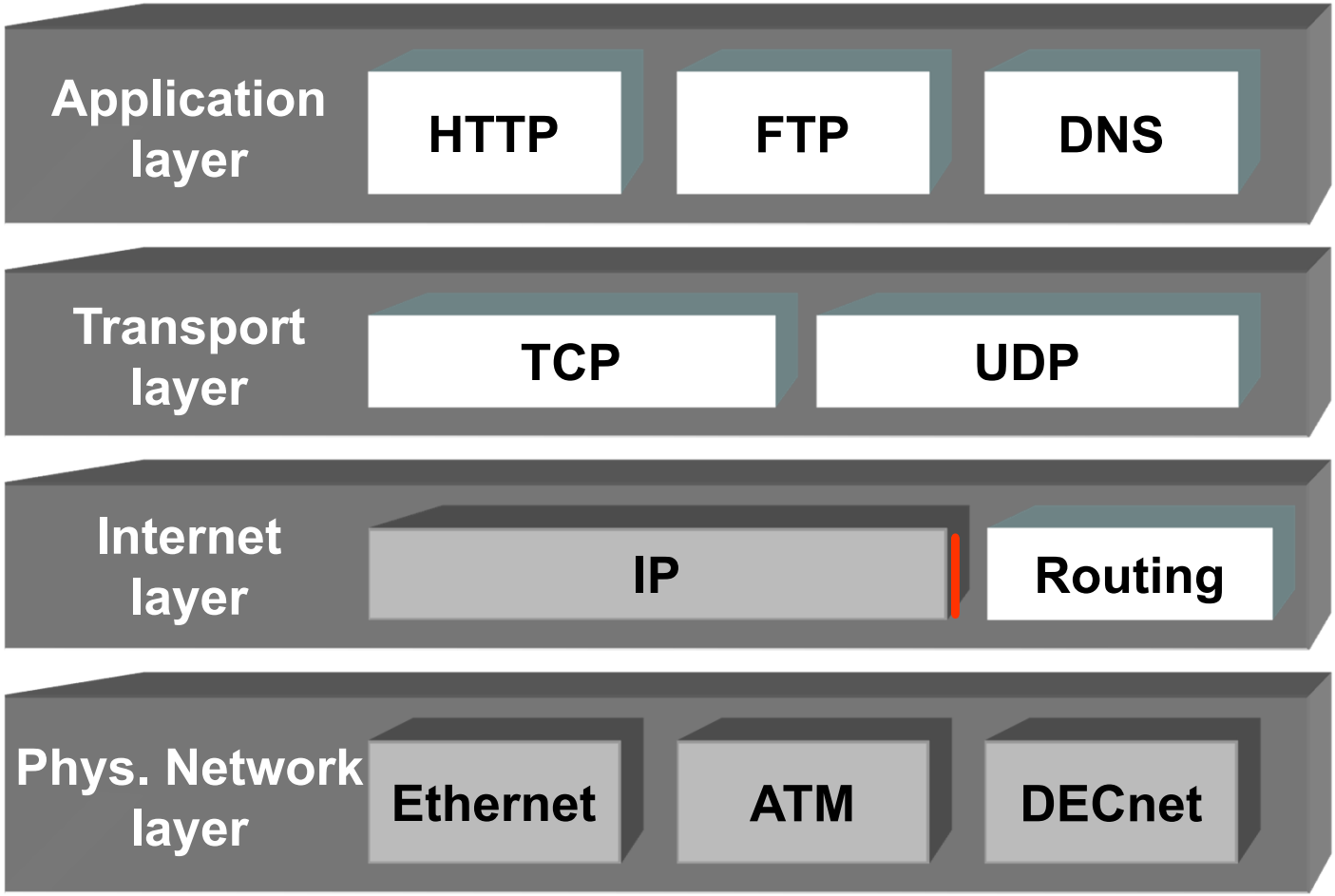
TECHNISCHE UNIVERSITÄT DARMSTADT

- Quick Recap of Internet Routing Architecture
  - Distance Vector Routing
  - Link State Routing
- Software Defined Networking
  - How does OpenFlow work?
  - OpenFlow is not enough
- SDN App example
- A Helpful Analogy

Quick Recap of Internet Routing Architecture

IP Protocol Stack

TECHNISCHE UNIVERSITÄT DARMSTADT



Routing vs. forwarding

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Routing (algorithm):**

  A successive exchange of connectivity information between routers. Each router builds its own routing table based on collected information.

- **Forwarding (process):**

  A switch- or router-*local* process which forwards packets towards the destination using the information given in the local routing table.

Routing algorithm

TECHNISCHE UNIVERSITÄT DARMSTADT

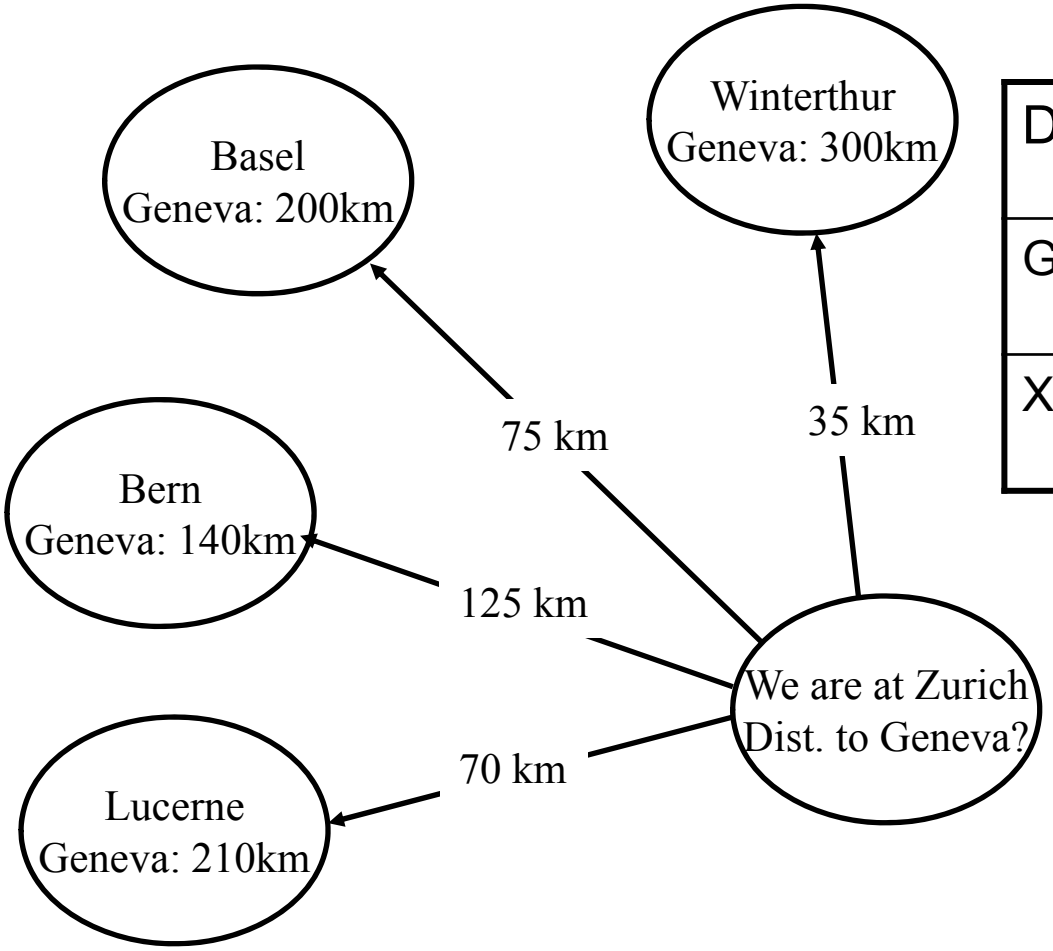
- A *distributed algorithm* executed among the routers which builds the routing tables. Path selection can be based on different metrics:
  - Quantative: #hops, bandwidth, available capacity, delay, jitter,…
  - Others: Policy, utilization, revenue maximization, politics,…
- Design and evaluation criteria:
  - Scalability of algorithm. How will *route information packets* (i.e. overhead) scale with an increased number of routers? Computational complexity?
  - Time to a common converged state.
  - Stability and robustness against errors and partial information
- Two important classes of routing algorithms
  - *Distance Vector* (also called Bellman-Ford or Ford-Fulkerson)
  - *Link State*

Richard Bellman: *On Routing Problem*, in Quarterly of Applied Mathematics, 16(1), pp.87-90, 1958.

Lestor R. Ford jr., D. R. Fulkerson: *Flows in Networks*, Princeton University Press, 1962.

Distance Vector Routing

Distance Vector Routing: Basic Idea



Dest	LU	BE	BS	W	opt
GE	280	265	275	335	BE
XY					

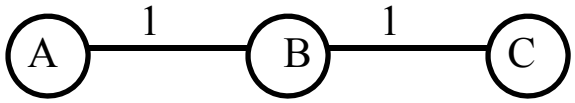
Distance Vector for GE as seen by the Zurich router: (GE, 265)

Distance Vector Routing - Description

TECHNISCHE UNIVERSITÄT DARMSTADT

- Each router reports a list of (directly or indirectly) *reachable destinations* and the *routing metric* ("distance vector") to its neighbors
- Each router updates its internal tables according to the information received. If a *shorter distance* to a destination is received, this is recorded in the table.
- The distance vector is sent *periodically* or when the routing table is changed (e.g. interval 30 seconds)
- Packets containing distance vectors are called *routing updates.*

Count-to-infinity Problem

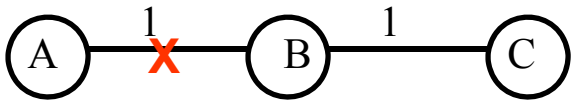


Node A		
Destination	Distance	Next node
B	1	B
C	2	B

Node B		
Destination	Distance	Next node
A	1	A
C	1	C

Noce C		
Destination	Distance	Next node
B	1	B
A	2	B

Count-to-infinity Problem

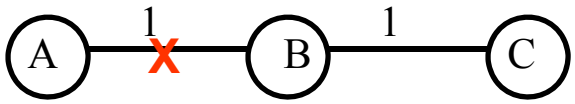


Node A		
Destination	Distance	Next node
B	1	B
C	2	B

Node B		
Destination	Distance	Next node
A	1	A
C	1	C

Node C		
Destination	Distance	Next node
B	1	B
A	2	B

Count-to-infinity Problem

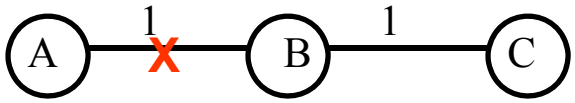


Node A		
Destination	Distance	Next node
B	N.E.	
C	N.E.	

Node B		
Destination	Distance	Next node
A	1	A
C	1	C

Node C		
Destination	Distance	Next node
B	1	B
A	2	B

Count-to-infinity Problem

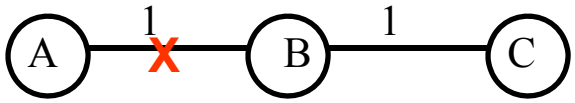


Node A		
Destination	Distance	Next node
B	N.E.	
C	N.E.	

Node B		
Destination	Distance	Next node
A	N.E.	
C	1	C

Node C		
Destination	Distance	Next node
B	1	B
A	2	B

Count-to-infinity Problem

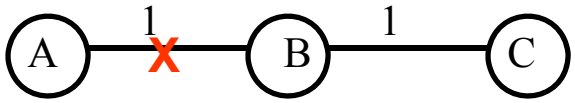


Node A		
Destination	Distance	Next node
B	N.E.	
C	N.E.	

Node B		
Destination	Distance	Next node
A	3	C
C	1	C

Node C		
Destination	Distance	Next node
B	1	B
A	2	B

Count-to-infinity Problem

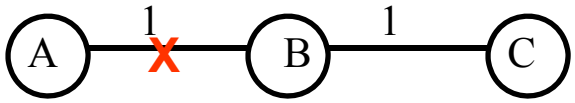


Node A		
Destination	Distance	Next node
B	N.E.	
C	N.E.	

Node B		
Destination	Distance	Next node
A	3	C
C	1	C

Node C		
Destination	Distance	Next node
B	1	B
A	4	B

Count-to-infinity Problem

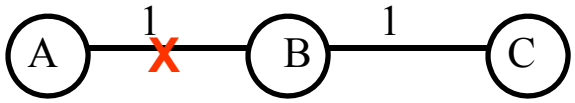


Node A		
Destination	Distance	Next node
B	N.E.	
C	N.E.	

Node B		
Destination	Distance	Next node
A	5	C
C	1	C

Node C		
Destination	Distance	Next node
B	1	B
A	4	B

Count-to-infinity Problem



Node A		
Destination	Distance	Next node
B	N.E.	
C	N.E.	

Node B		
Destination	Distance	Next node
A	5	C
C	1	C

Node C		
Destination	Distance	Next node
B	1	B
A	6	B

Bad news travel slow[ly]

Fixes

TECHNISCHE UNIVERSITÄT DARMSTADT

- Define infinity as finite
  - Maximum hop count is 15, ≥16 means infinite
- Split horizon
  - Never advertise a route out of the interface through which you learned it.
- Poison reverse
  - Advertise invalid routes as *unreachable*
- Split horizon with poison reverse
  - Once you learn of a route through an interface, advertise it as unreachable back through that same interface.
- Hold-down timer
- Report the entire path

Link State Routing

Link State Routing: Basic idea

TECHNISCHE UNIVERSITÄT DARMSTADT

- Each router compiles a list of *directly* connected neighbors with associated metric
- Each router participates in *flooding* these lists
- Convergence: With time, each router will get the *full topology* of the network.
- Routers compute the best route from a source (or themselves) to a destination using Dijkstra's Shortest Path First (SPF) algorithm

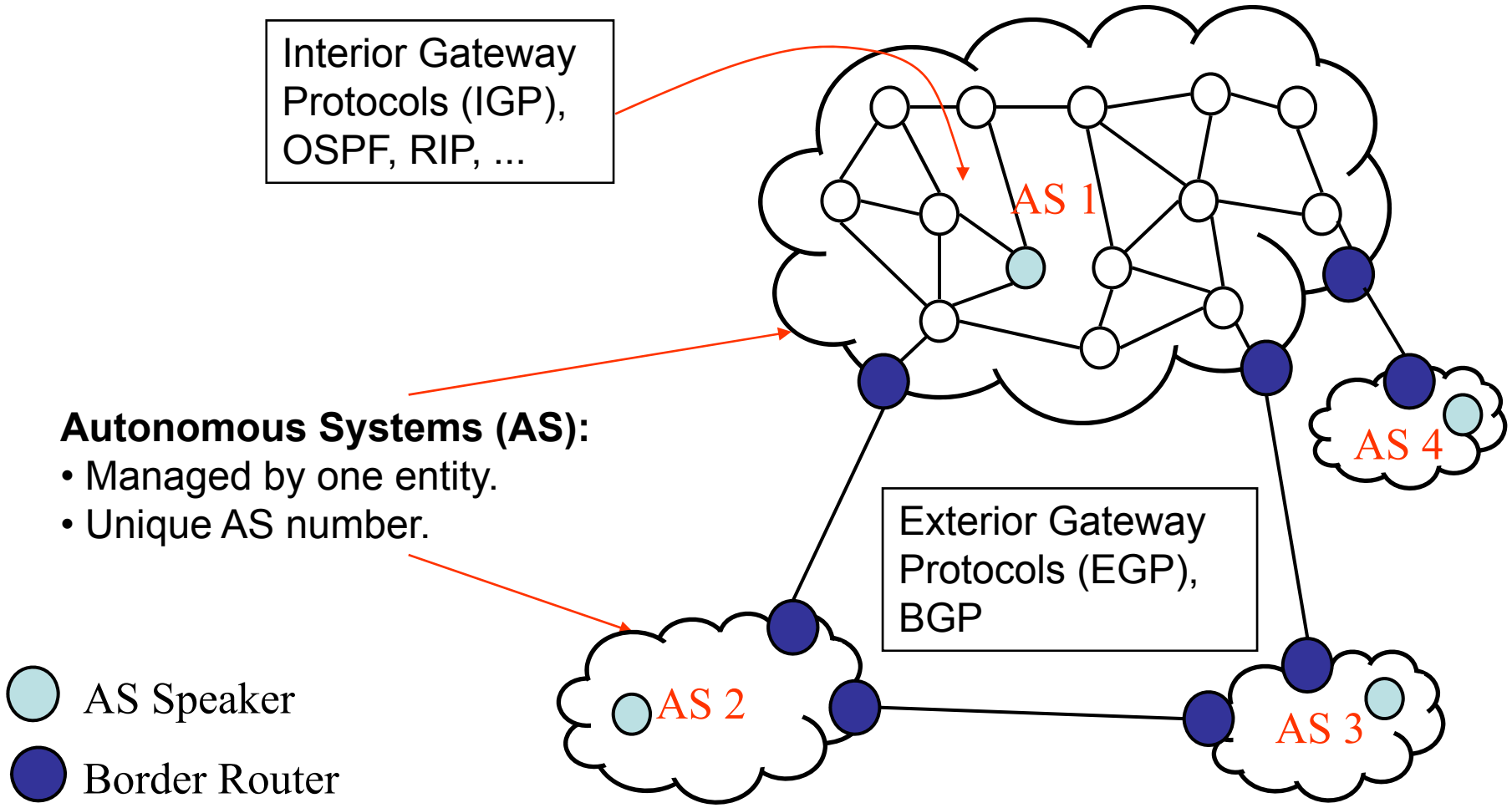
Motivation for *hierarchical routing*

TECHNISCHE UNIVERSITÄT DARMSTADT

- Scalability
  - Both algorithms (DV, LS) have poor scalability properties (memory and computational complexity).
  - DV also has some problem with number and size of routing updates.

- Administration may need more facilities, e.g.
  - Local routing policies
  - Specific metrics (hops, delay, traffic load, cost, …)
  - Medium-term traffic management
  - Different levels of trust (own routers / foreign routers)

Hierarchical routing domains, AS



Internet intra-domain routing protocols

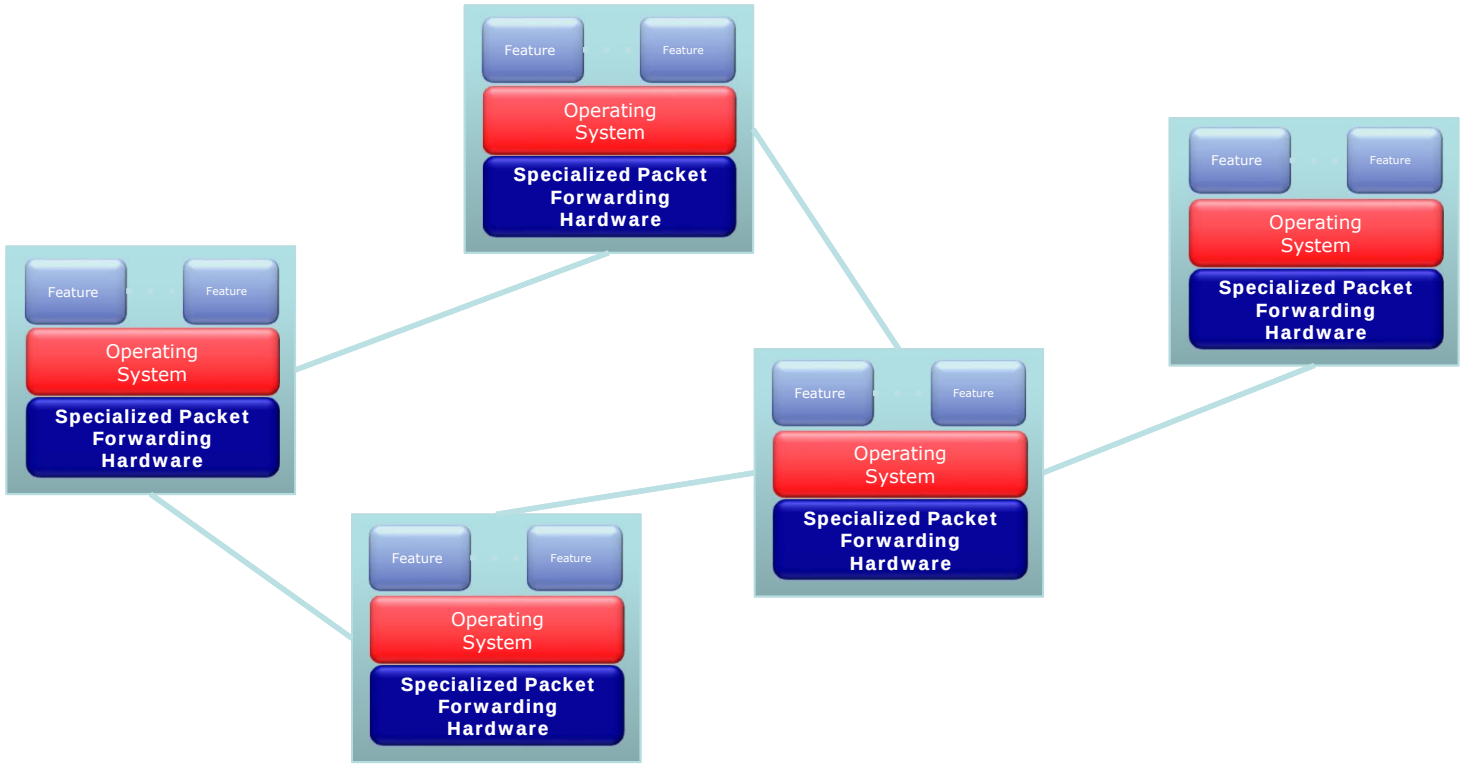
TECHNISCHE UNIVERSITÄT DARMSTADT

- Distance-Vector-type:
  - Routing Information Protocol (**RIP**), RFC 1058, 2453
- Link-State-type
  - Open Shortest Path First (**OSPF**), RFC 2328
  - Intermediate System-to-Intermediate System (IS-IS), an OSI protocol supported by most routers

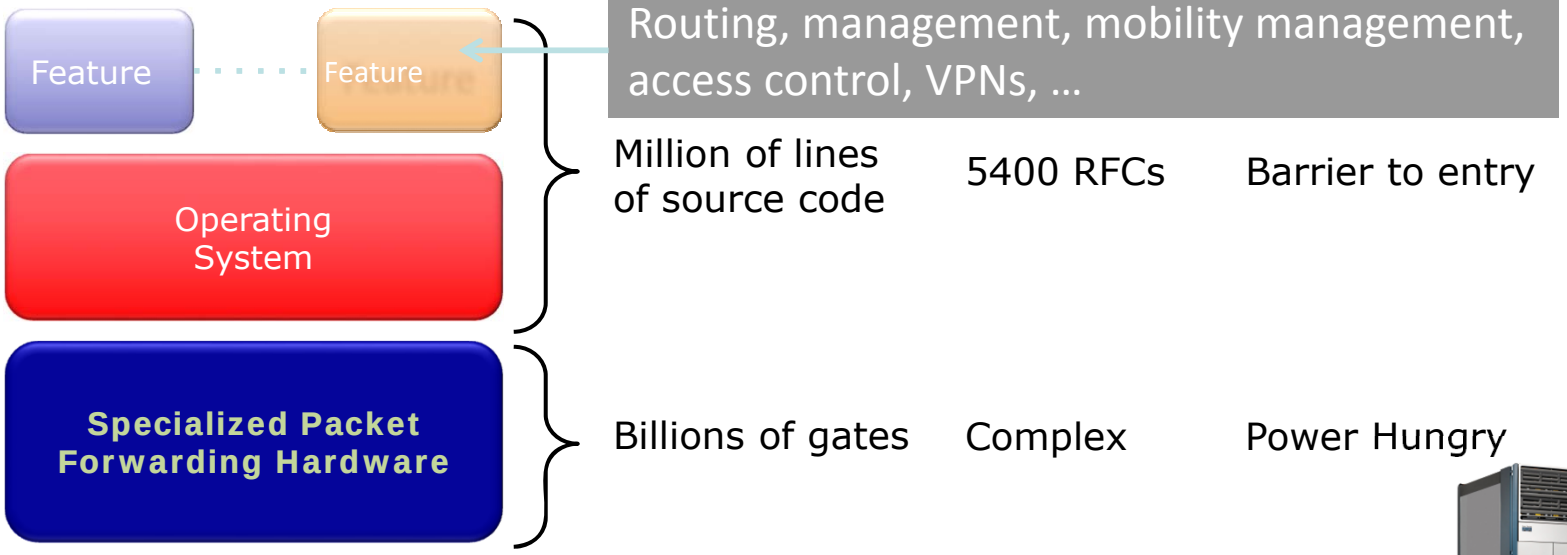
Classical network architecture

TECHNISCHE UNIVERSITÄT DARMSTADT

- Distributed control plane
- Distributed routing protocols: OSPF, IS-IS, BGP, etc.



The Networking Industry (2007)



Closed, vertically integrated, boated, complex, proprietary

Many complex functions baked into the infrastructure

OSPF, BGP, multicast, differentiated services,
Traffic Engineering, NAT, firewalls, MPLS, redundant layers, …

Little ability for non-telco network operators to get what they want

Functionality defined by standards, put in hardware, deployed on nodes

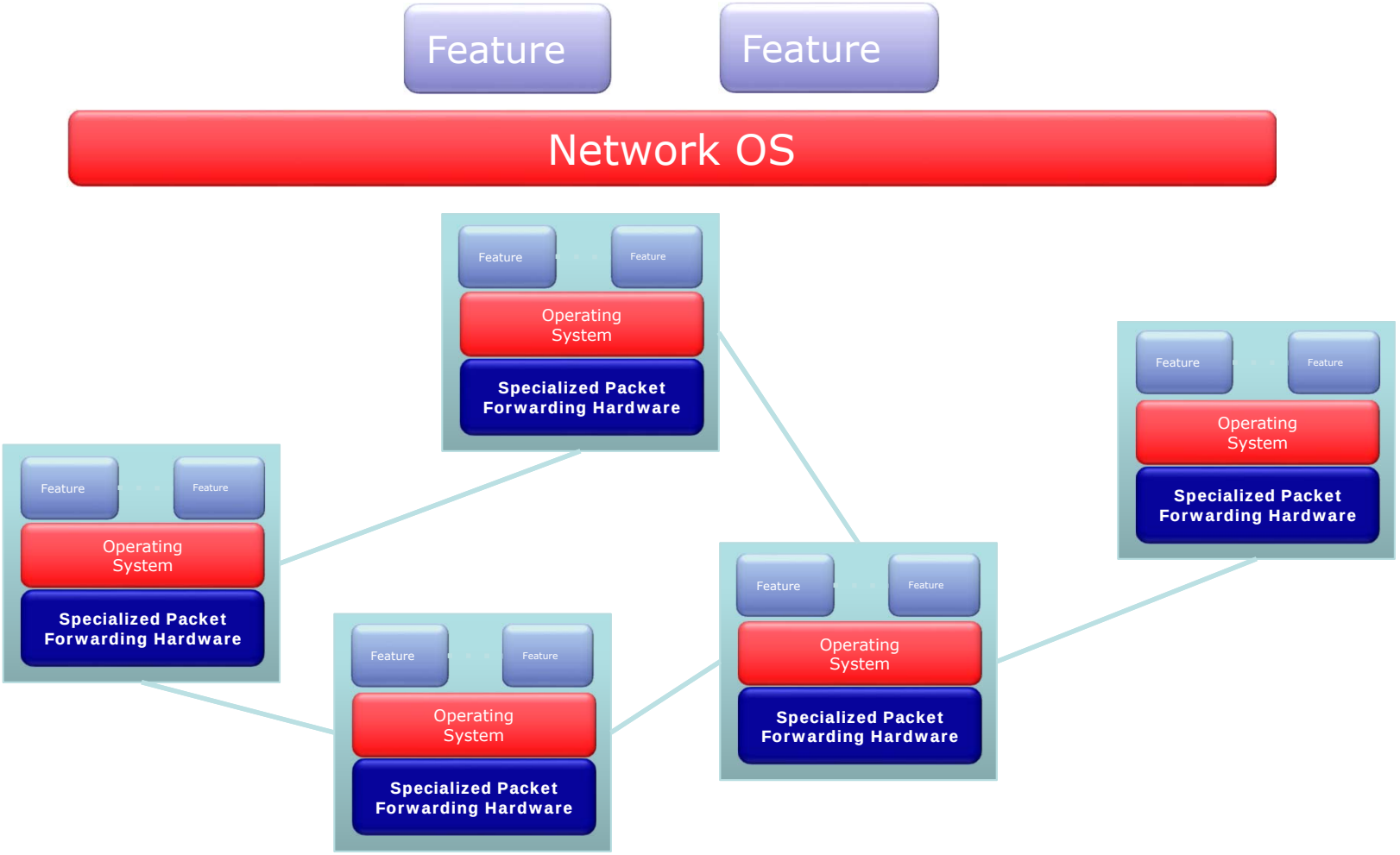
Software Defined Networking

SDN

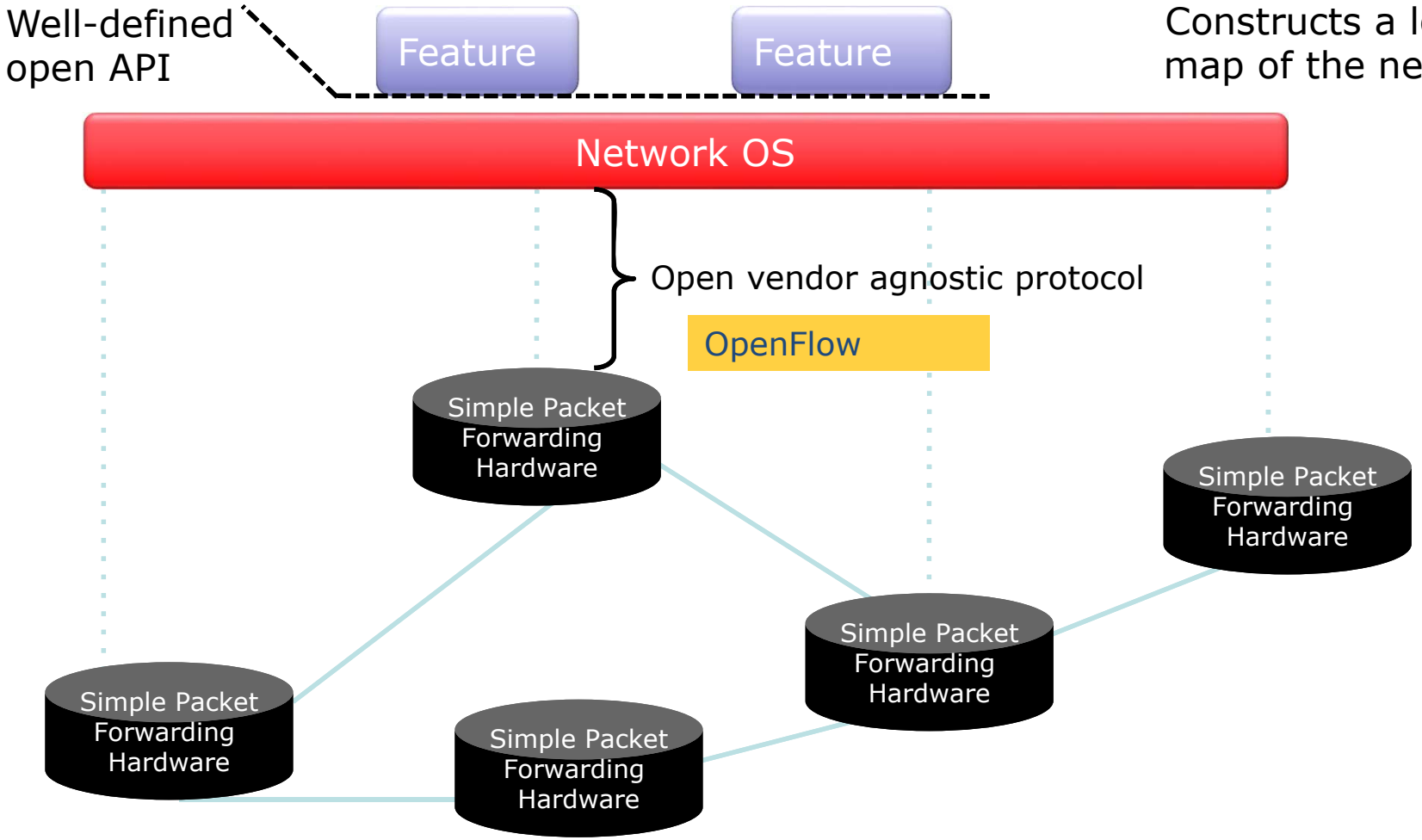
TECHNISCHE UNIVERSITÄT DARMSTADT

- Possible definitions:
  - SDN is a new network architecture:
    - that's makes it easier to program networks.
    - with the core idea that software remotely controls network hardware.
    - …

From Vertically Integrated to …



Software Defined Network



Network OS

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Network OS:** distributed system that creates a consistent, up-to-date network view
  - Runs on servers (controllers) in the network

- Uses an open protocol to:
  - Get state information **from** forwarding elements
  - Give control directives **to** forwarding elements

OpenFlow

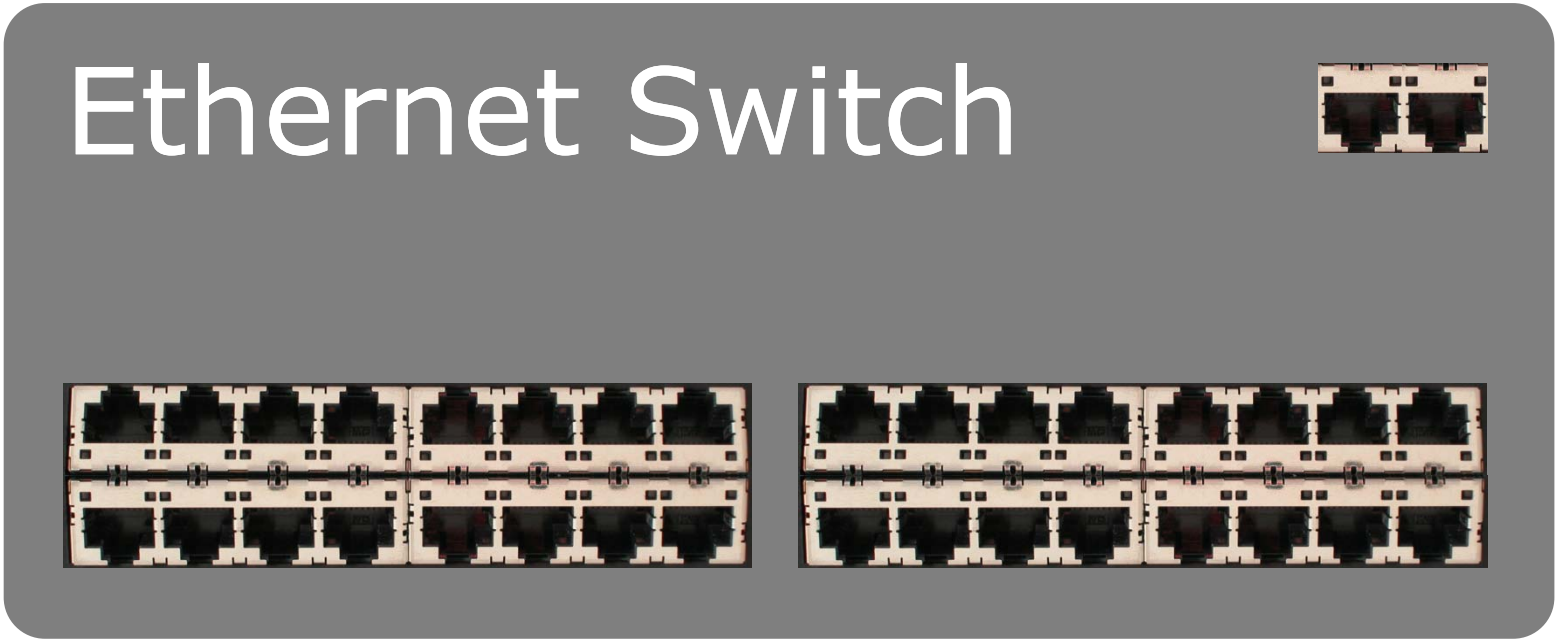
TECHNISCHE UNIVERSITÄT DARMSTADT

- OpenFlow
  - is a protocol for remotely controlling the forwarding table of a switch or router
  - is one element of SDN

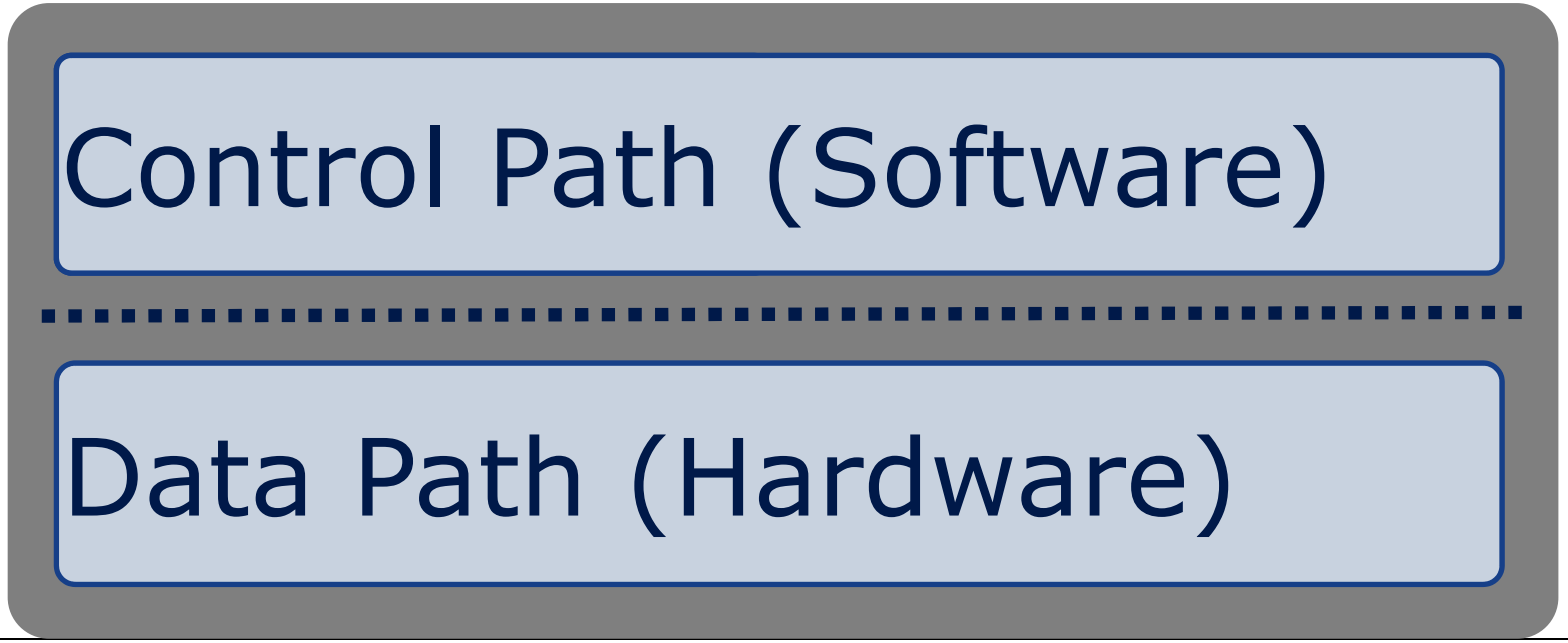
How does OpenFlow work?

Ethernet Switch

TECHNISCHE UNIVERSITÄT DARMSTADT

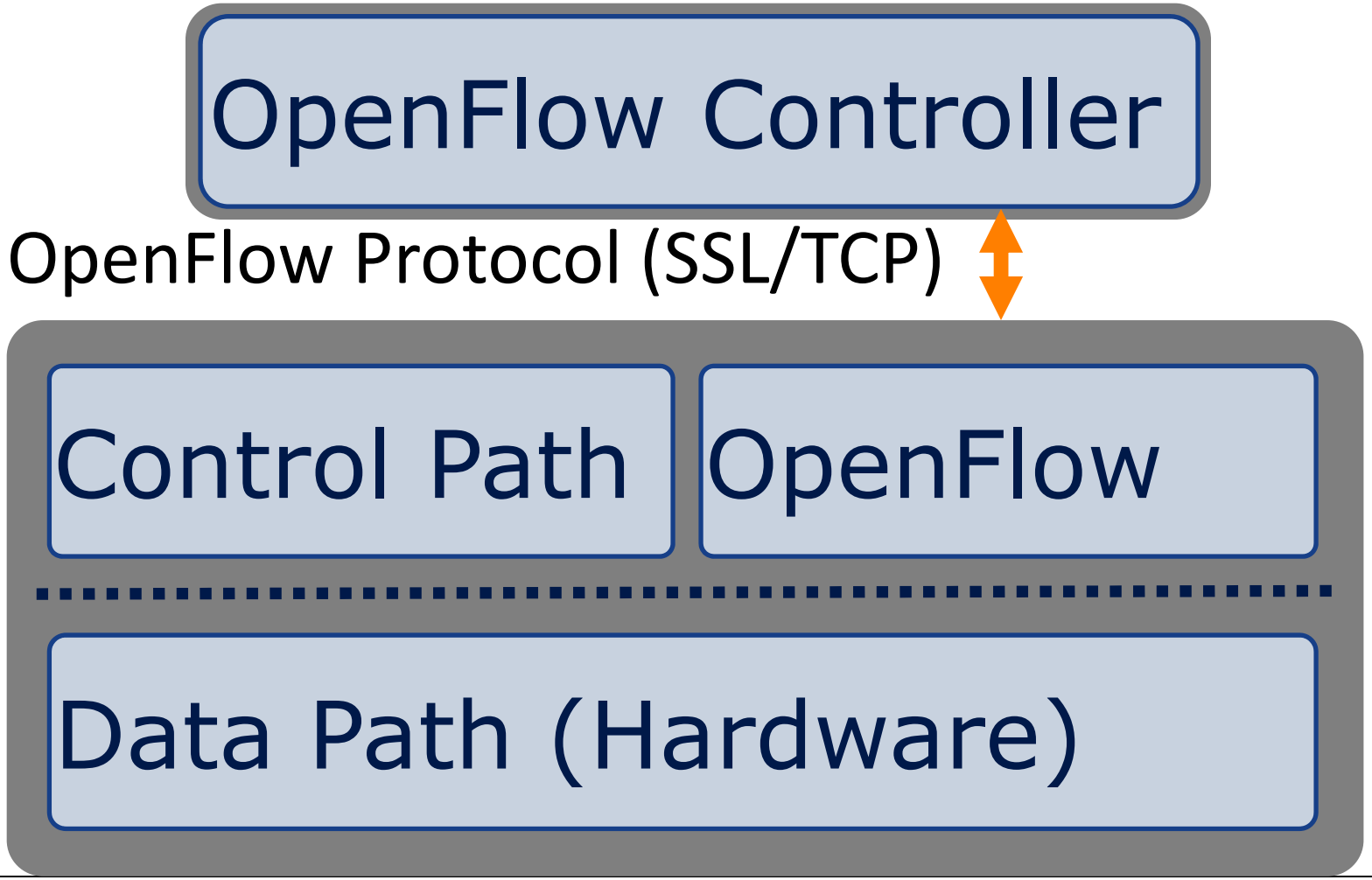


Control Path vs. Data Path

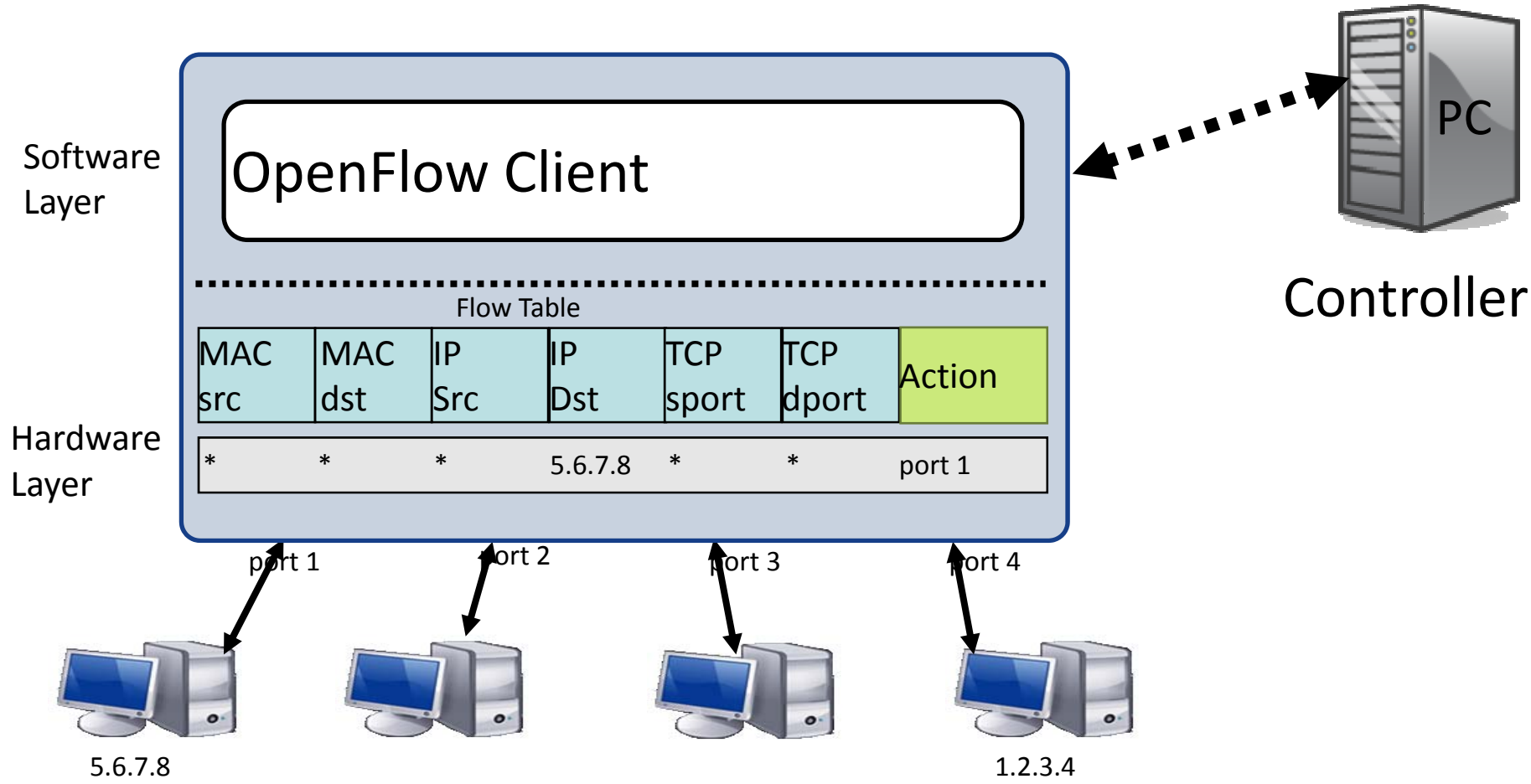


OpenFlow Protocol

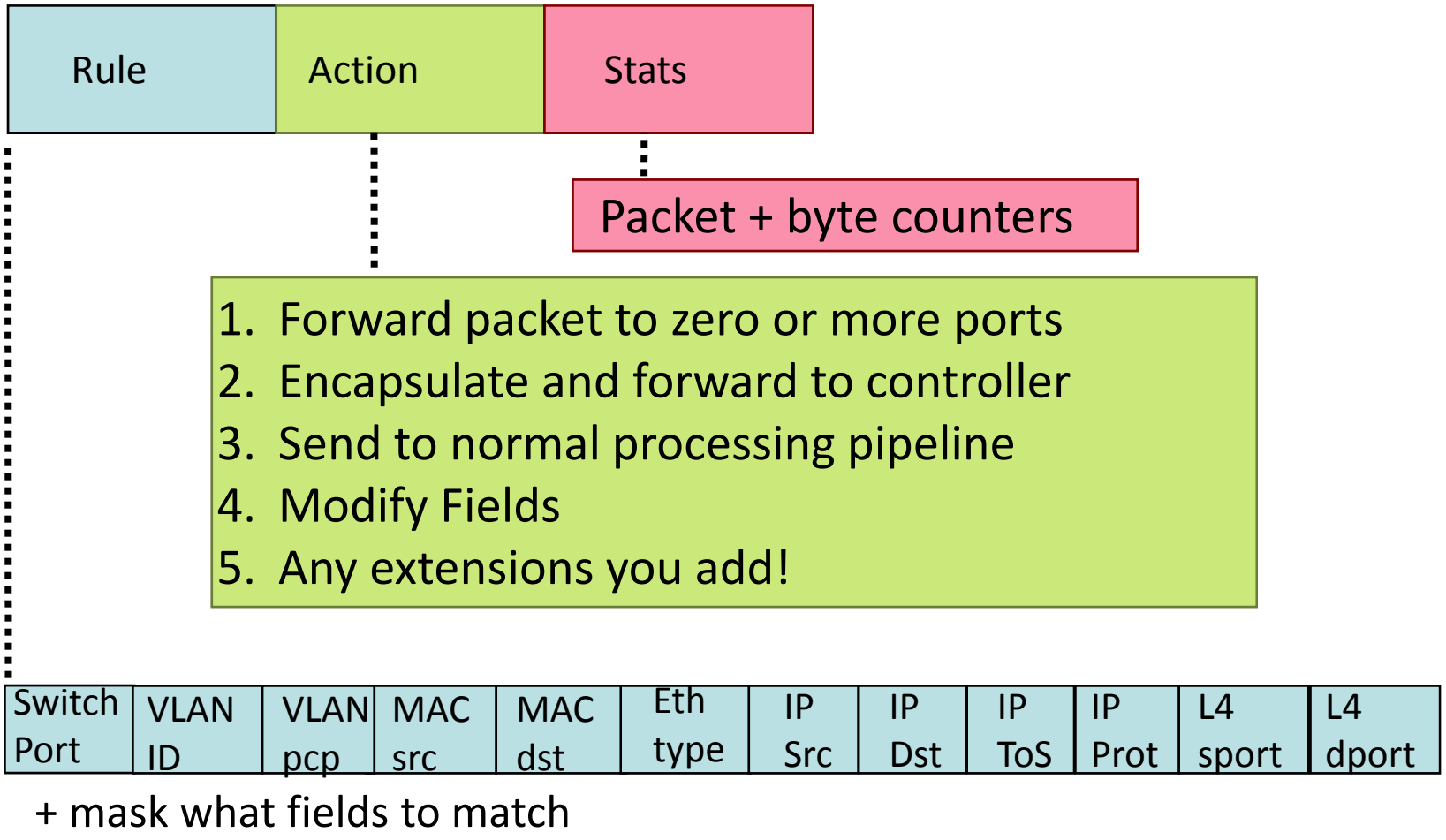
TECHNISCHE UNIVERSITÄT DARMSTADT



OpenFlow Example



OpenFlow Basics: Flow Table Entries



Examples

TECHNISCHE UNIVERSITÄT DARMSTADT

Switching

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	00:1f:..	*	*	*	*	*	*	*	port6

Flow Switching

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
port3	00:20..	00:1f..	0800	vlan1	1.2.3.4	5.6.7.8	4	17264	80	port6

Firewall

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	*	*	*	22	drop

Examples

TECHNISCHE UNIVERSITÄT DARMSTADT

Routing

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	5.6.7.8	*	*	*	port6

VLAN Switching

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	00:1f..	*	vlan1	*	*	*	*	*	port6, port7, port9

Secure Channel

TECHNISCHE UNIVERSITÄT DARMSTADT

- A secure channel from switch to controller is needed
  - SSL connection, site-specific key
    - Provides encryption and authentication
- Security is important
  - Opening the interface opens up new possibilities for attacks
- A controller discovery protocol is needed
  - When a new switch is installed it initially has an empty fwding table and does not know how to forward packets.
  - The discovery protocol broadcasts the presence of a switch so that a controller can establish an association with the switch and configure its fwd table.
- Encapsulate packets for controller
- Send link/port state to controller

Main Concepts of Architecture

TECHNISCHE UNIVERSITÄT DARMSTADT

- Separate data from control
  - A standard protocol between data and control
- Define a generalized flow table
  - Very flexible and generalized flow abstraction
  - Open up layers1-7
- Open control API
  - For control and management applications
- Virtualization of the data and control plane
- Backward compatible
  - Though allows completely new header

OpenFlow is not enough

OpenFlow is not enough…

TECHNISCHE UNIVERSITÄT DARMSTADT

- Adds the ability to modify, experiment…
- But still harder than it should be to add features to a network
- Effectively assembly programming or an ISA (instruction set architecture)

[OpenFlow is just a forwarding table management protocol]

TECHNISCHE UNIVERSITÄT DARMSTADT

SDN App example

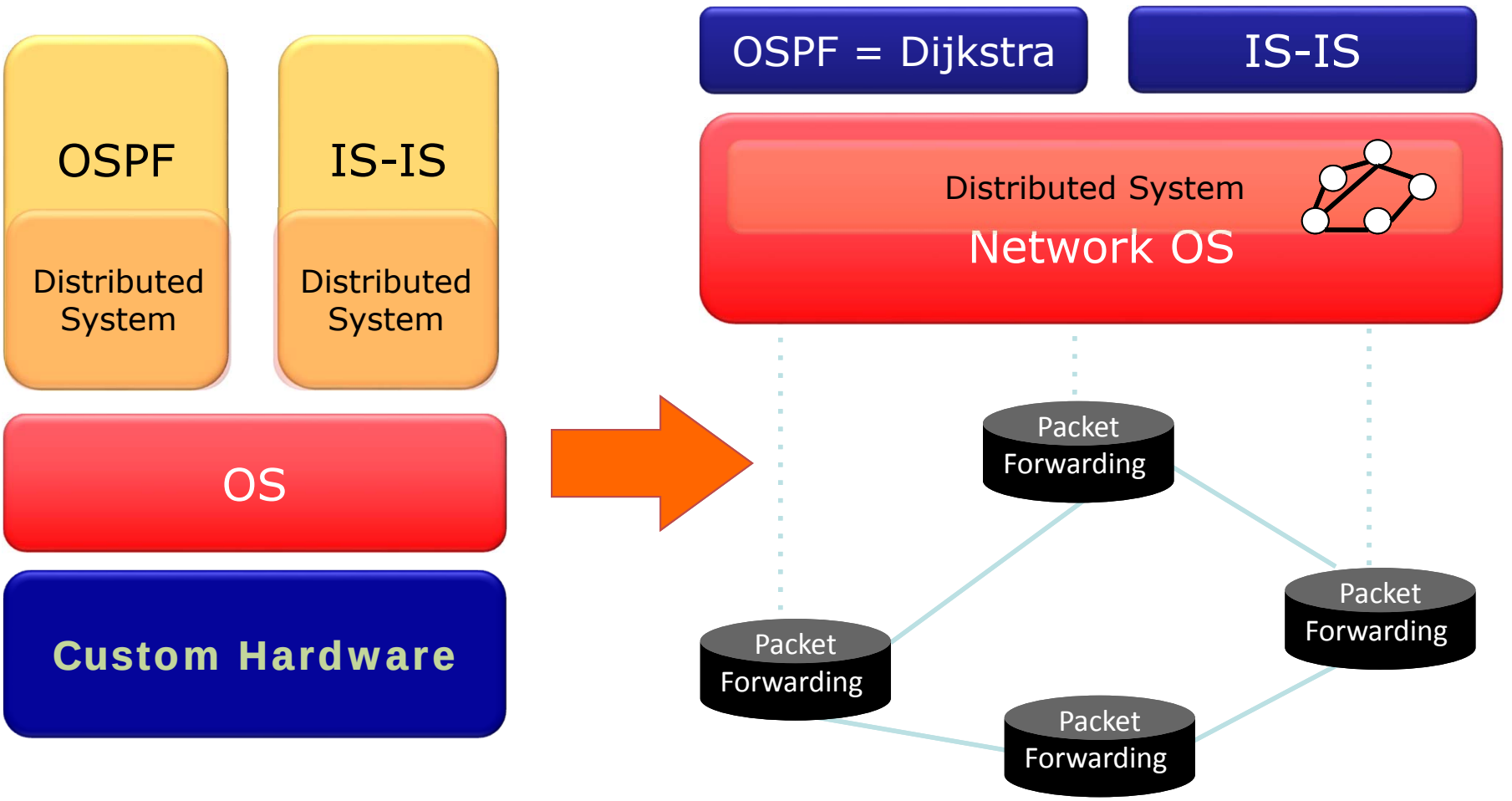
OSPF and Dijkstra

SDN App example

TECHNISCHE UNIVERSITÄT DARMSTADT

- OSPF
  - RFC 2328: **245 pages**
- Distributed Protocol
  - Builds consistent, up-to-date map of the network: **101 pages**
- Dijkstra's Algorithm
  - Operates on map: **4 pages**

Example


OSPF
IS-IS
Distributed System
Distributed System
OS
Custom Hardware
OSPF = Dijkstra
IS-IS
Distributed System
Network OS
Packet Forwarding
Packet Forwarding
Packet Forwarding
Packet Forwarding

Other SDN Use Cases

TECHNISCHE UNIVERSITÄT DARMSTADT

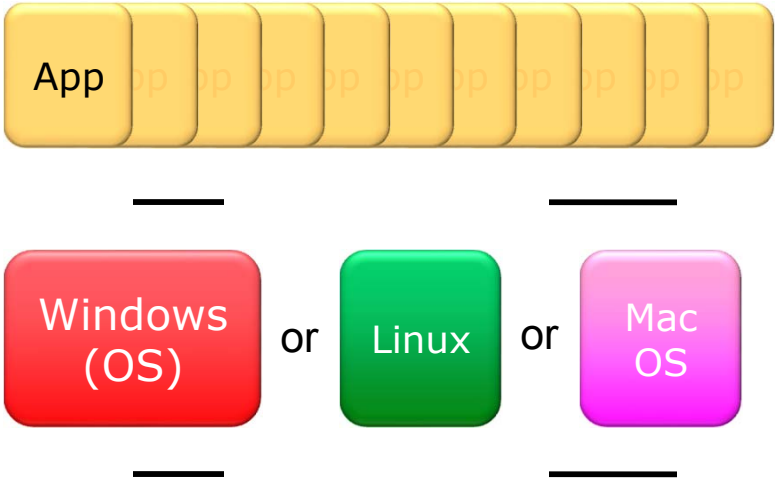
- Energy conservation, routing, and management in data centers
- Seamless use of diverse wireless networks
- Network based load balancing
- Traffic engineering
- Slicing and scalable remote control/management of home networks
- Experimentation with new approaches and protocols using selected production traffic
- Run virtual shadow network for traffic analysis and re-configuration
- And many more …

See http://www.openflow.org/videos/

A Helpful Analogy

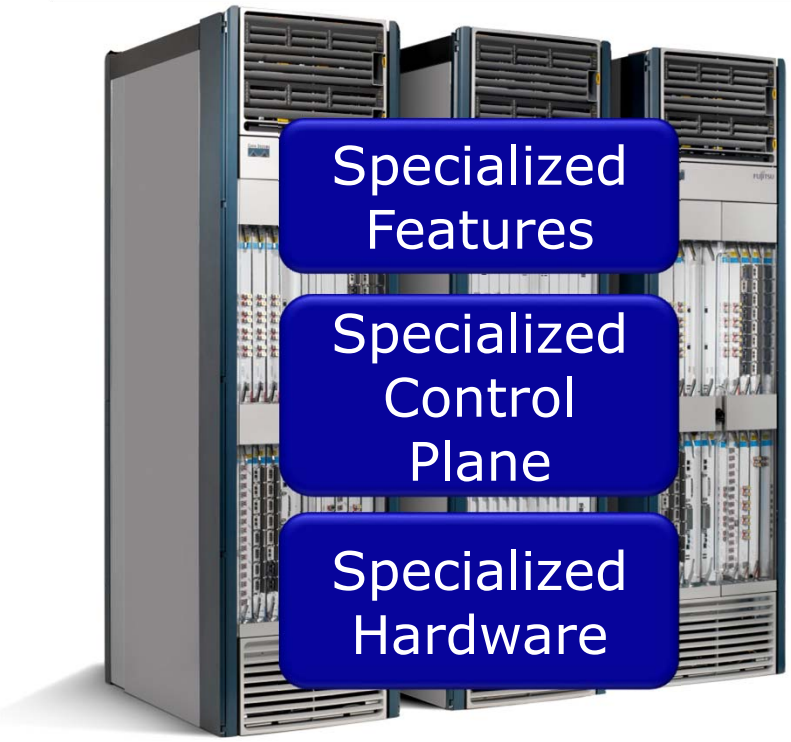
A Helpful Analogy

Mainframe industry in the 1980s: Vertically integrated
Closed, proprietary
Slow innovation
Small industry

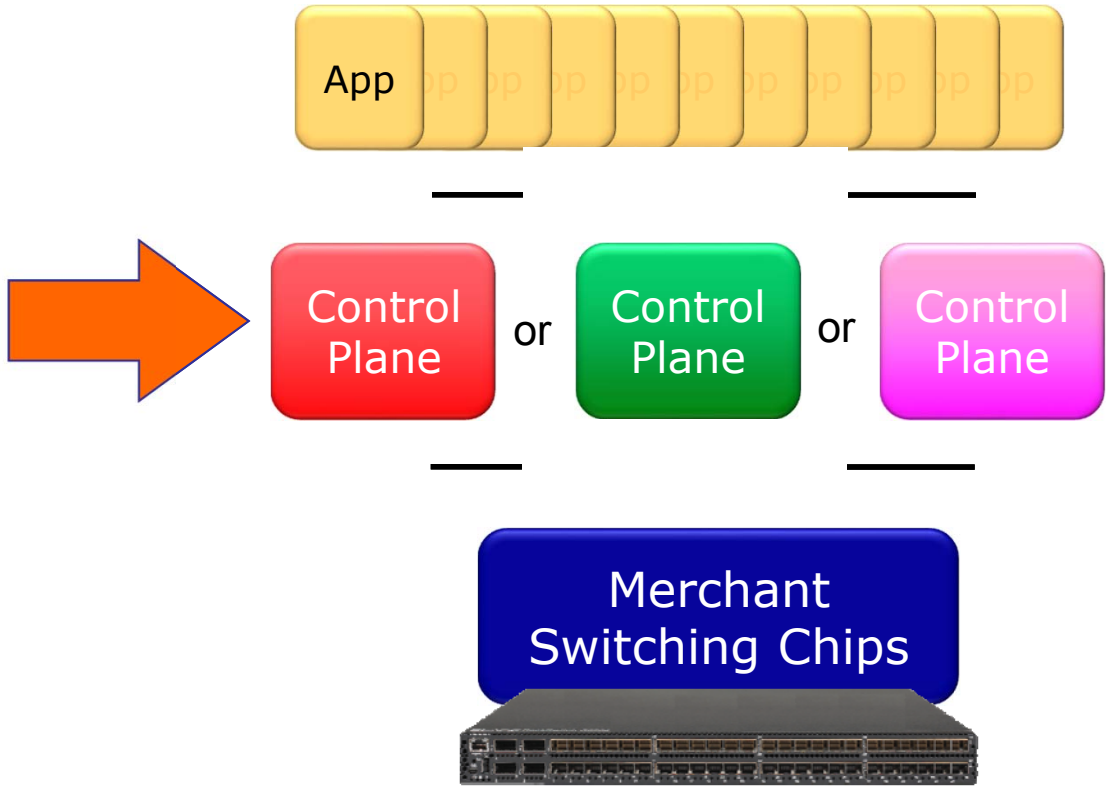


Horizontal
Open interfaces
Rapid innovation
Huge industry

A Helpful Analogy



Networking industry in 2007: Vertically integrated Closed, proprietary Slow innovation



Horizontal Open interfaces Rapid innovation