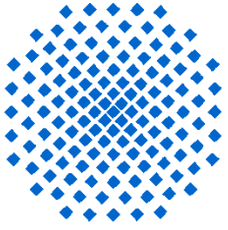


Universität Stuttgart

Institute of Parallel and
Distributed Systems (IPVS)

Universitätsstraße 38
D-70569 Stuttgart

Tutorial: Software-defined Networking

Part 1: Motivation and Introduction

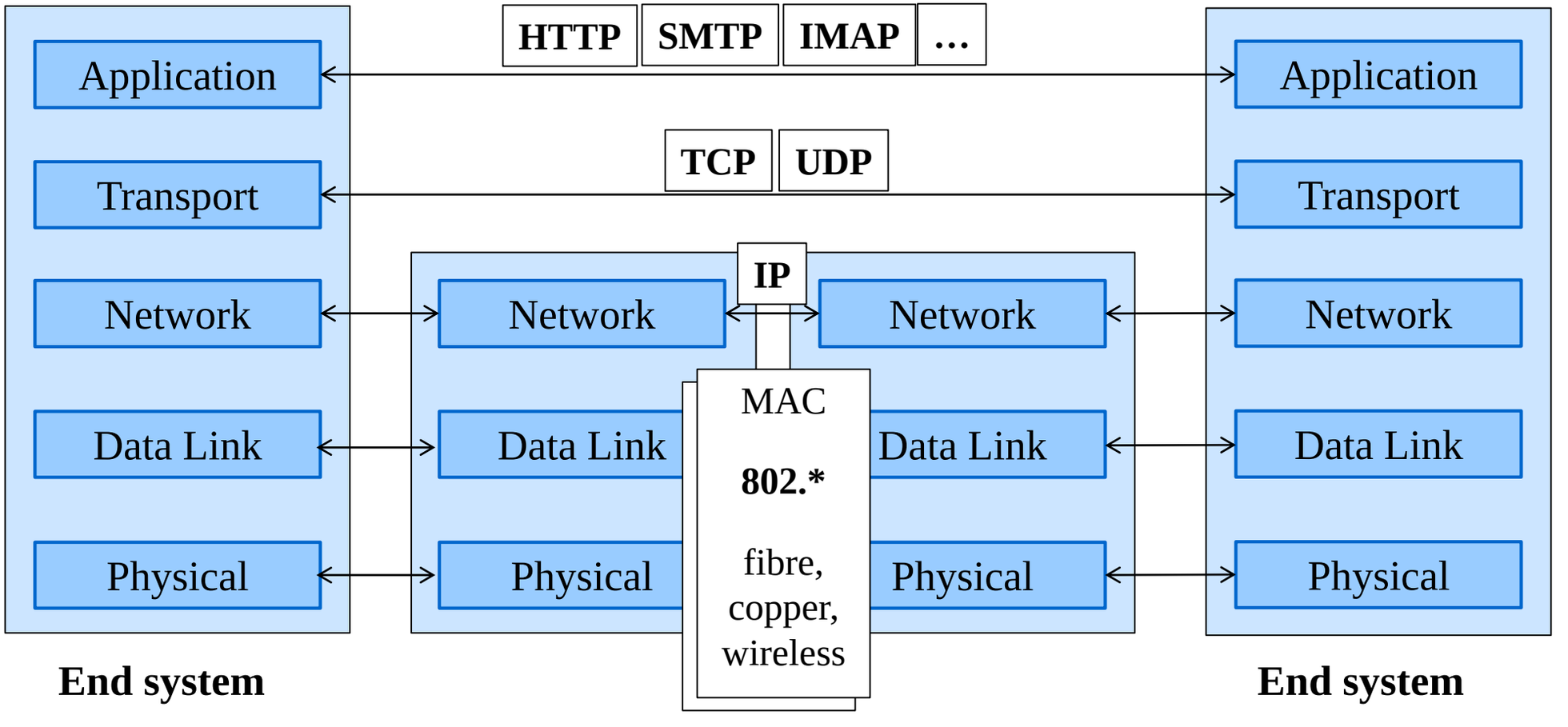
Frank Dürr

Overview

- Motivation of Software-defined Networking
  - Problems and limitations of current networking infrastructures
  - Motivating examples
- Introduction to Software-Defined Networking (SDN)

The Internet Protocol Stack: A Success Story

The basic architecture remained the same for decades.



Constant Evolution in Networking

- Basic network architecture stayed the same
- However, there is a constant evolution going on

➔ New networked systems with new requirements appear constantly

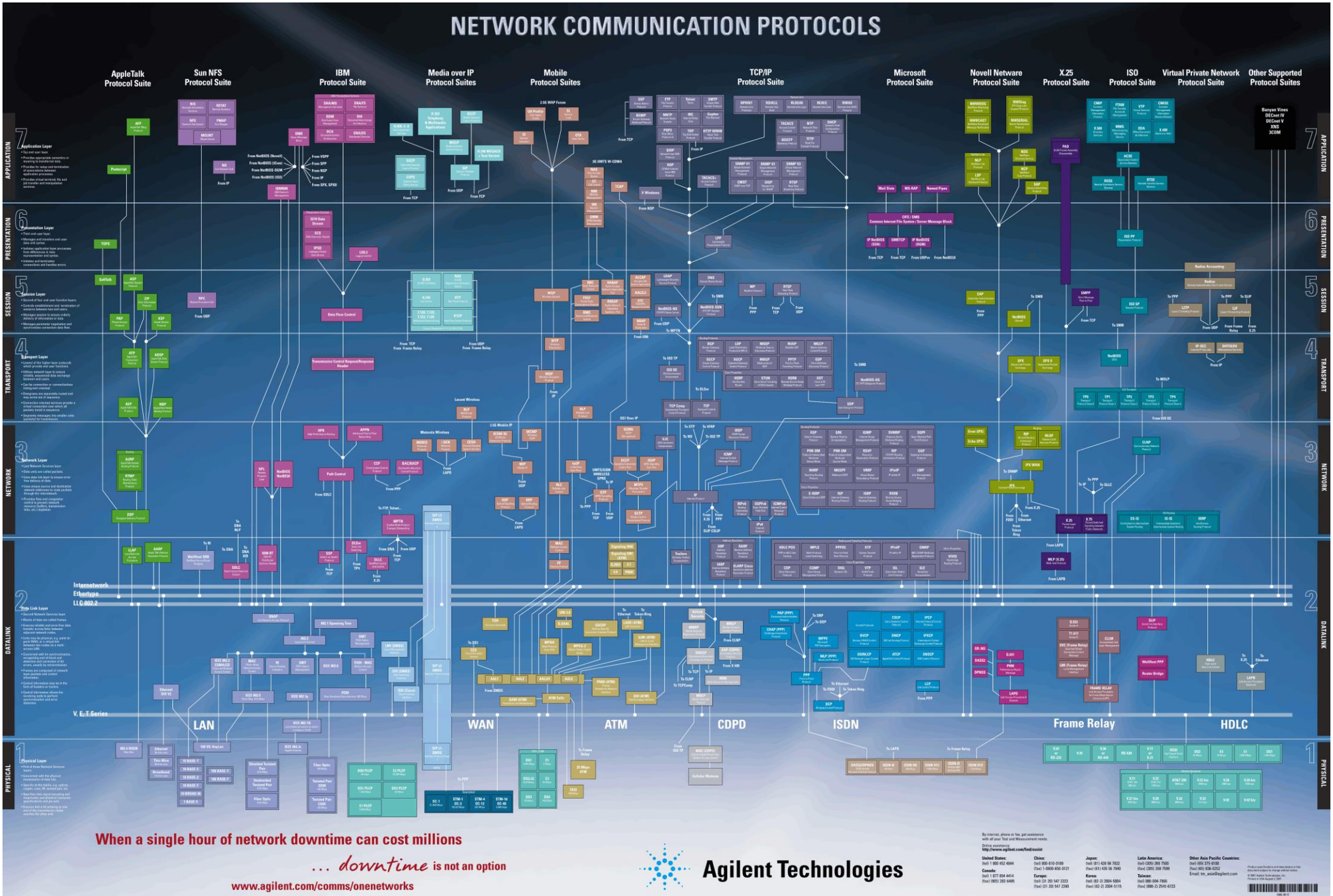
Growing Number of Networking Scenarios (1)

- Unicast, multicast, anycast, information-centric networking, …
  - Different addressing and routing concepts
- Heterogeneous Quality of Service (QoS)
  - Real-time and low-latency applications
    - Soft-real time requirements: Web search, shopping systems, social networks, instant messaging
    - Strict real-time requirements: Networked control systems
  - Delay-tolerant applications
    - Backups, email, file transfer
  - High throughput applications
    - “Big data” storage and processing (e.g. map/reduce jobs)

Growing Number of Networking Scenarios (2)

- Virtual networks (VLANs)
  - Isolation of communication flows (e.g., in a cloud data center)
- Connection of local networks via wide-area network
  - VPNs tunneling of flows between networks
- Mobility support
  - Transparent forwarding of flows to mobile device
  - Migration of virtual machines in or across data centers
- And more …

Consequence: Constantly Increasing Number of Network Protocols and Standards



Challenge and Problems

Challenge: Implement networking infrastructure supporting all of these functions, requirements, and protocols

Problems of today's networking infrastructure:

- Limited flexibility
- Separation of network and application

Problem: Limited Flexibility (1)

- Switches and routers are closed “black” boxes
- Support standard protocols and proprietary protocols of manufacturer
- No easy changes without support of manufacturer

➔ Network protocols seem to be “hard-coded“

Problem: Limited Flexibility (2)

And even if the switch/router hardware and software is open:
Adding new network protocols and functions is hard

- Have you ever written a Linux kernel module?
  - Compare this to programming a user-space application
- Have you ever used VERILOG?
  - Compare this to C/C++, Java, Python, …
- Have you ever targeted an embedded device?
  - Compare this to writing code for a server

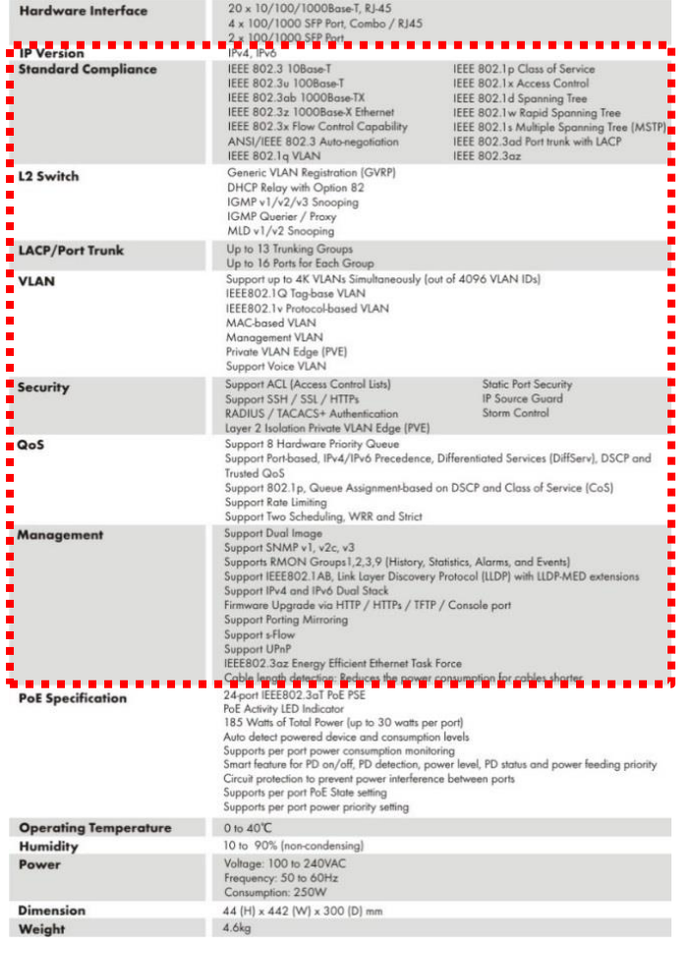
WRT54L

NetFPGA (http://netfpga.org/)

Problem: Increasing Switch/Router Complexity

Switches and router implement large set of complex protocols

- Even redundant protocols like multiple routing protocols
- Even protocols that are not needed by the application / customer
- Customer pays for hardware resources and functionality that he might not need
  - … and still cannot easily implement his own protocols



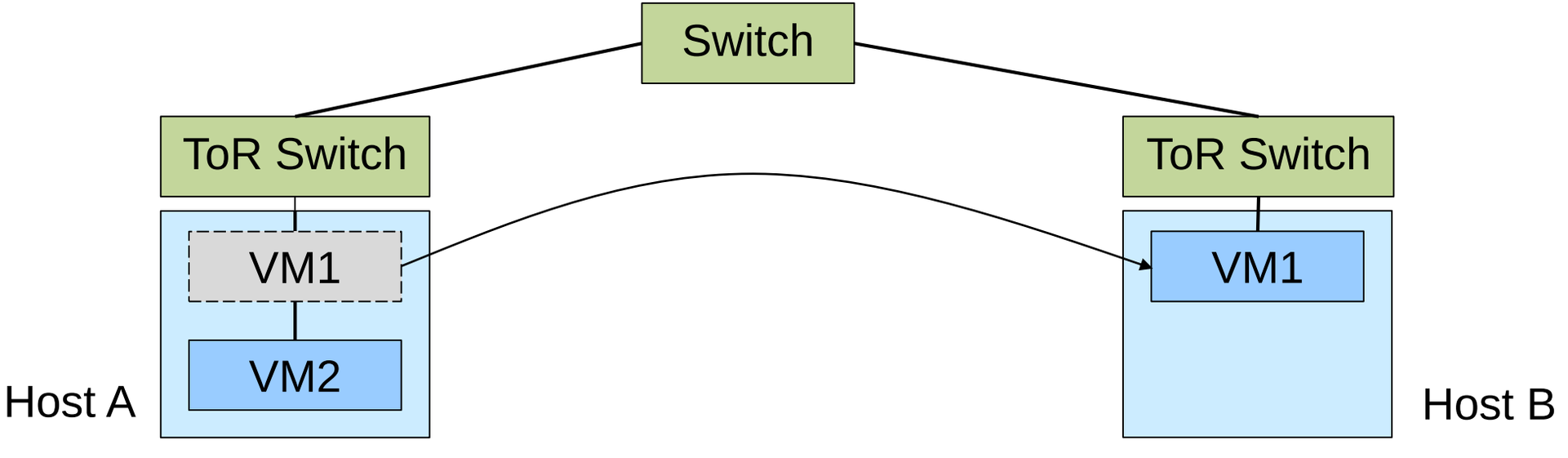
Hardware Interface	20 x 10/100/1000Base-T, RJ-45 4 x 100/1000 SFP Port, Combo / RJ45 2 x 100/1000 SFP Port	
IP Version	IPv4, IPv6	
Standard Compliance	IEEE 802.3 10Base-T IEEE 802.3u 100Base-T IEEE 802.3ab 1000Base-TX IEEE 802.3z 1000Base-X Ethernet IEEE 802.3x Flow Control Capability ANSI/IEEE 802.3 Auto-negotiation IEEE 802.1q VLAN	IEEE 802.1p Class of Service IEEE 802.1x Access Control IEEE 802.1d Spanning Tree IEEE 802.1w Rapid Spanning Tree IEEE 802.1s Multiple Spanning Tree (MSTP) IEEE 802.3ad Port trunk with LACP IEEE 802.3az
L2 Switch	Generic VLAN Registration (GVRP) DHCP Relay with Option 82 IGMP v1/v2/v3 Snooping IGMP Querier / Proxy MLD v1/v2 Snooping	
LACP/Port Trunk	Up to 13 Trunking Groups Up to 16 Ports for Each Group	
VLAN	Support up to 4K VLANs Simultaneously (out of 4096 VLAN IDs) IEEE802.1Q Tag-base VLAN IEEE802.1v Protocol-based VLAN MAC-based VLAN Management VLAN Private VLAN Edge (PVE) Support Voice VLAN	
Security	Support ACL (Access Control Lists) Support SSH / SSL / HTTPs RADIUS / TACACS+ Authentication Layer 2 Isolation Private VLAN Edge (PVE)	Static Port Security IP Source Guard Storm Control
QoS	Support 8 Hardware Priority Queue Support Port-based, IPv4/IPv6 Precedence, Differentiated Services (DiffServ), DSCP and Trusted QoS Support 802.1p, Queue Assignment-based on DSCP and Class of Service (CoS) Support Rate Limiting Support Two Scheduling, WRR and Strict	
Management	Support Dual Image Support SNMP v1, v2c, v3 Supports RMON Groups1,2,3,9 (History, Statistics, Alarms, and Events) Support IEEE802.1AB, Link Layer Discovery Protocol (LLDP) with LLDP-MED extensions Support IPv4 and IPv6 Dual Stack Firmware Upgrade via HTTP / HTTPs / TFTP / Console port Support Porting Mirroring Support s-Flow Support UPnP IEEE802.3az Energy Efficient Ethernet Task Force Cable length detection: Reduces the power consumption for cable shorter	
PoE Specification	24-port IEEE802.3aT PoE PSE PoE Activity LED Indicator 185 Watts of Total Power (up to 30 watts per port) Auto detect powered device and consumption levels Supports per port power consumption monitoring Smart feature for PD on/off, PD detection, power level, PD status and power feeding priority Circuit protection to prevent power interference between ports Supports per port PoE State setting Supports per port power priority setting	
Operating Temperature	0 to 40℃	
Humidity	10 to 90% (non-condensing)	
Power	Voltage: 100 to 240VAC Frequency: 50 to 60Hz Consumption: 250W	
Dimension	44 (H) x 442 (W) x 300 (D) mm	
Weight	4.6kg	

Problem: Separation of Network and Application

- Application view onto the network: **byte pipe (black box)**
- Network view onto the application: **load generator (black box)**
- **Integrated system view** would benefit application and network!
  - Higher performance of application
  - Higher efficiency and utilization of network

Example: VM Migration

- Virtual Machine moved from Host A to B
- Switches reactively find new location of host using ARP protocol
  - High load through broadcasts, slow reaction
- Why not let management application “program” the network proactively based on its knowledge of VM locations and network topology?



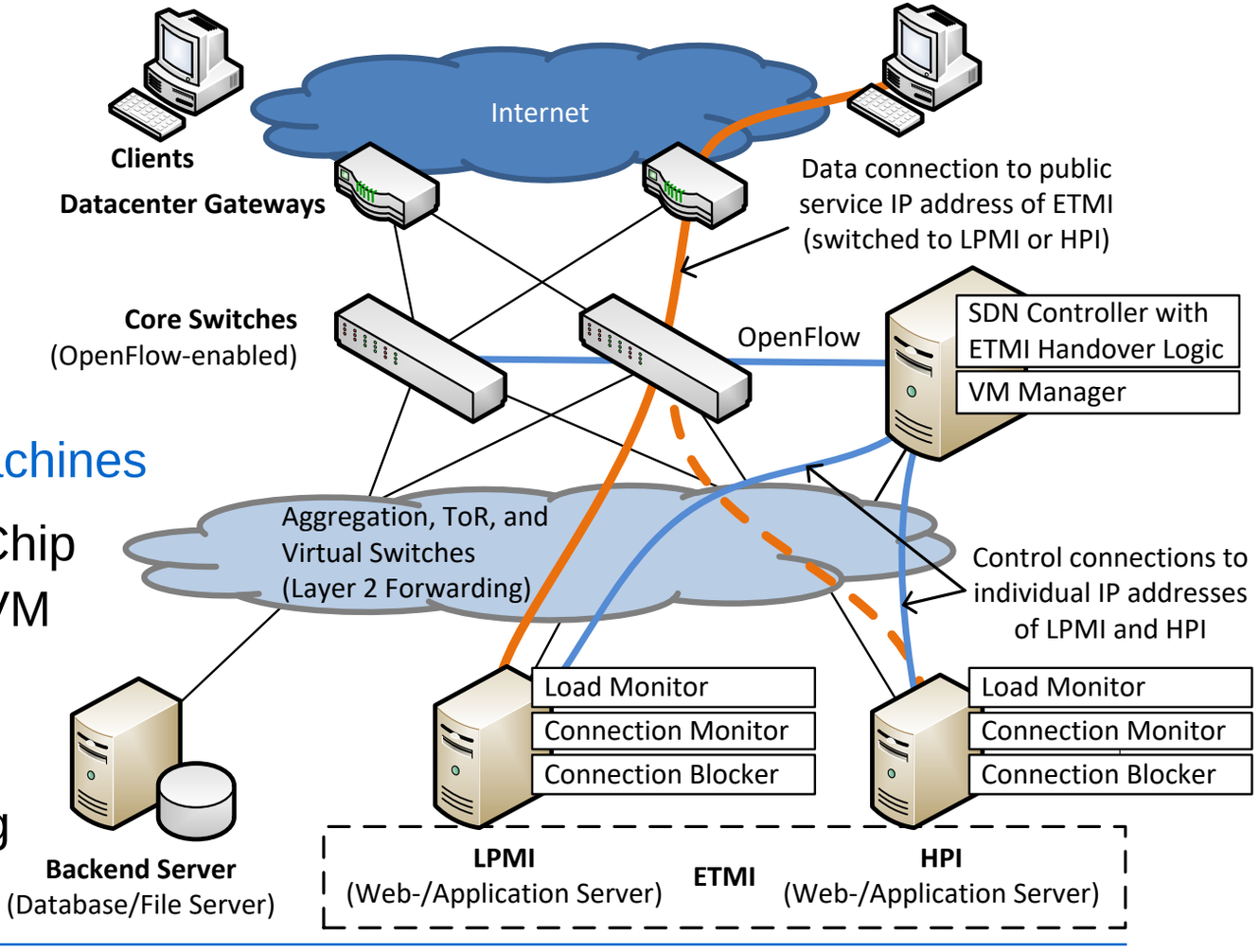
Example: Elastic Tandem Machines based on SDN

Goal: Energy-proportional machine for data centers

- Low-power consumption in idle mode and low load
- Scale-up to nominal (maximum) resources of VM

Approach: Elastic Tandem Machines

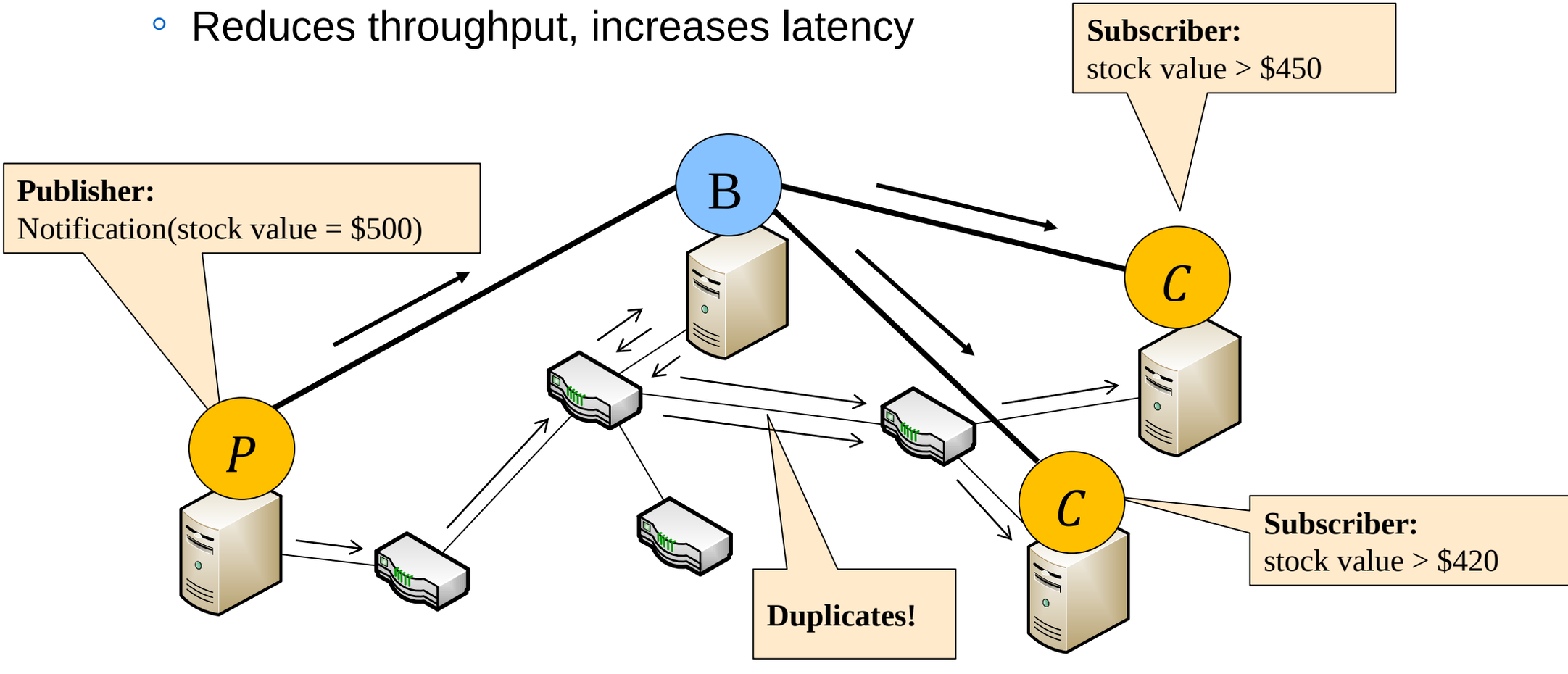
- Integration of System-on-a-Chip (SoC) machine and classic VM
- Seamless and transparent handover in network through Software-defined Networking



Frank Dürr: ***Improving the Efficiency of Cloud Infrastructures with Elastic Tandem Machines.***
In Proceedings of the 6th IEEE International Conference on Cloud Computing (Cloud 2013), pp. 91–98, Santa Clara, CA, USA, June 2013

Example: Publish/Subscribe

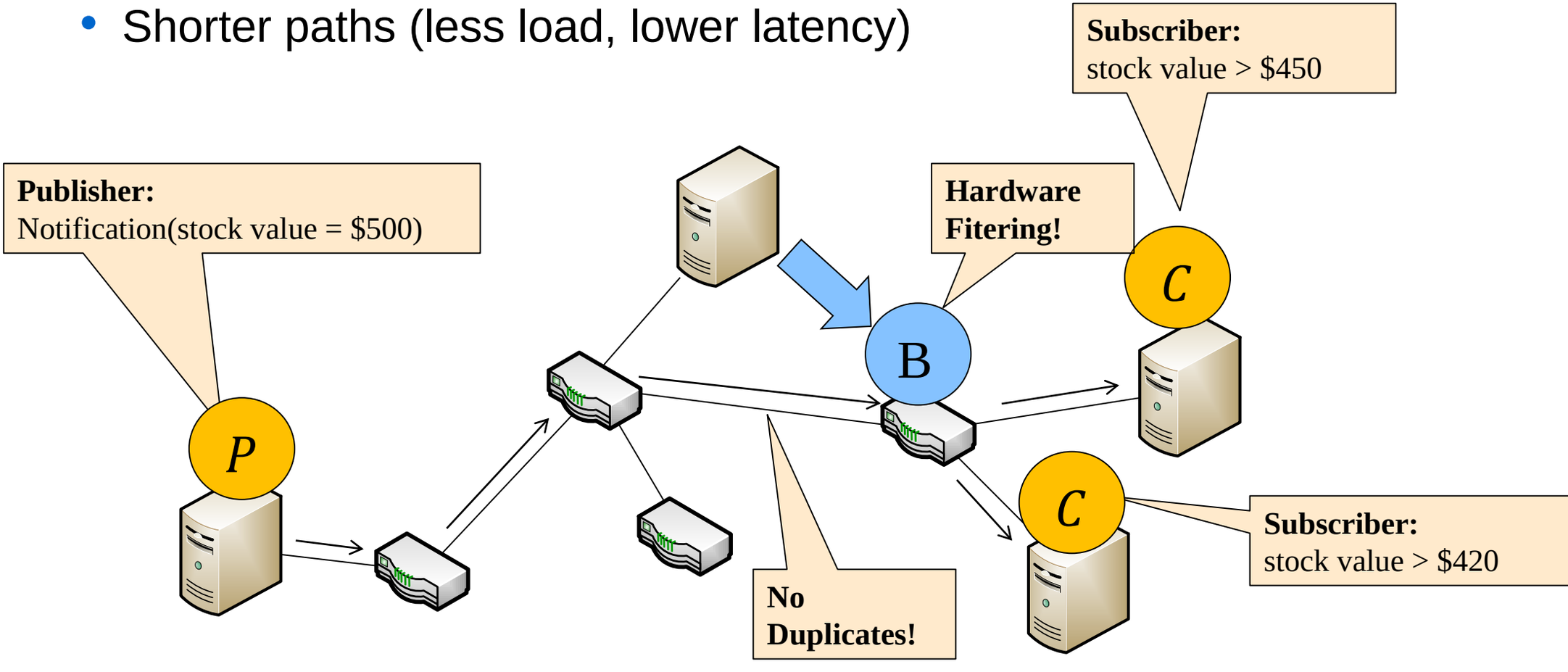
- Classic approach: Overlay networks
  - Messages are transmitted multiple times
  - Increased path length
- Forwarding implemented in software
  - Reduces throughput, increases latency



Example: Publish/Subscribe

With SDN

- Reduced number of messages
  (less network load, better scalability)
- Faster filtering in hardware (micro-seconds vs. milliseconds)
- Shorter paths (less load, lower latency)



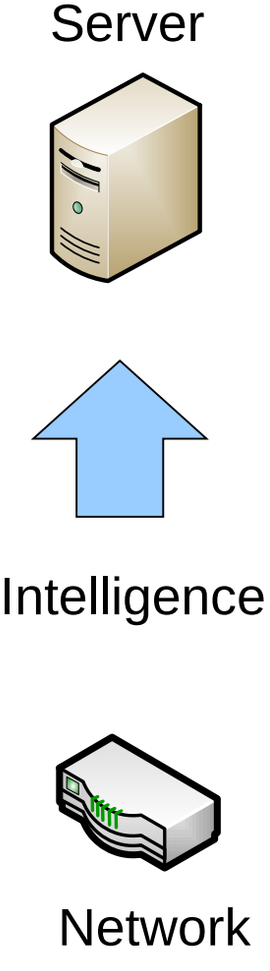
Benefits of SDN (1)

Software defined-networking leverages increased flexibility

- **Easy modification of the network control logic**
  - From “hard-coded” logic to exchangeable software
- **API to “program” the network**
  - Software (application) “defines” the network
- **High-level programming languages**
  - For implementation of logic
  - To benefit from powerful integrated developing environments

Benefits of SDN (2)

- **Reduced switch complexity**
  - Remove control logic from switch and host it on servers
- **Integrated system: application & network**
  - Global view onto the system
- **High performance of forwarding utilizing hardware switches**
  - Line-rate throughput
  - Micro-seconds forwarding latency
- **Reducing the complexity of implementing control logic**
  - Distribution transparency



Tremendous Industrial Interest

OPEN NETWORKING FOUNDATION

Open Network Foundation: www.opennetworking.org

OpenFlow

- > 100 members and many key players
- Manufacturer of switches, telecommunication operators
- Supporter and operator of data center

Tremendous Industrial Interest

SDN products and deployments already available

- Optimizing traffic in Google's backbone network
  - Going from < 40 % network utilization (industry standard) to nearly 100 %
  - Urs Hölzle's talk: https://www.youtube.com/watch?v=JMkvCBOMhno
- Open vSwitch
  - Prominent software switch
  - Developed by start-up Nicira (bought by VMware for $1.26 billion)
  - Now open source
- Hardware switches from major vendors

Overview

- Motivation of Software-defined Networking
  - Problems and limitations of current networking infrastructures
  - Motivating examples
- **Introduction to Software-Defined Networking (SDN)**

SDN in a Nutshell

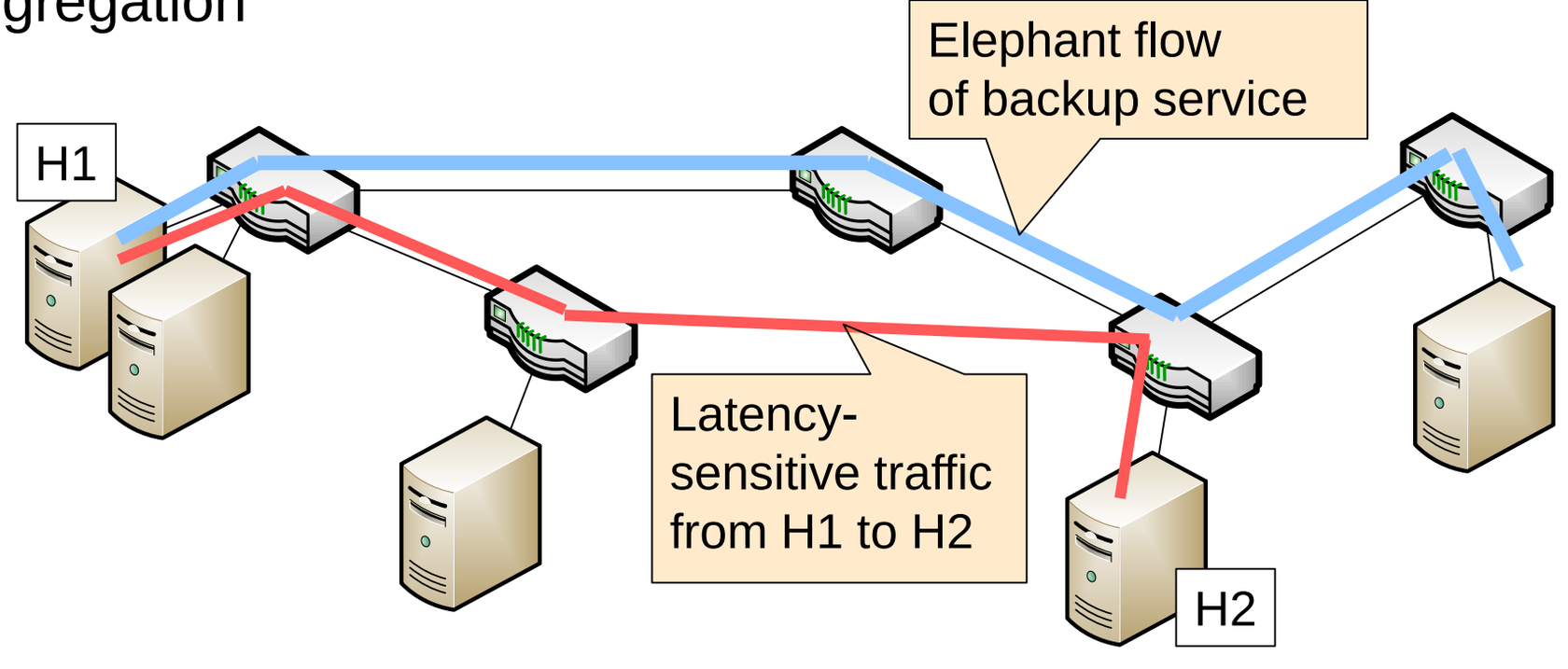
SDN is a paradigm to program networks at a high-level

"Think of it as a general language or an instruction set that lets me write a control program for the network rather than having to rewrite all of code on each individual router," *-Scott Shenker*

Flow-based Forwarding

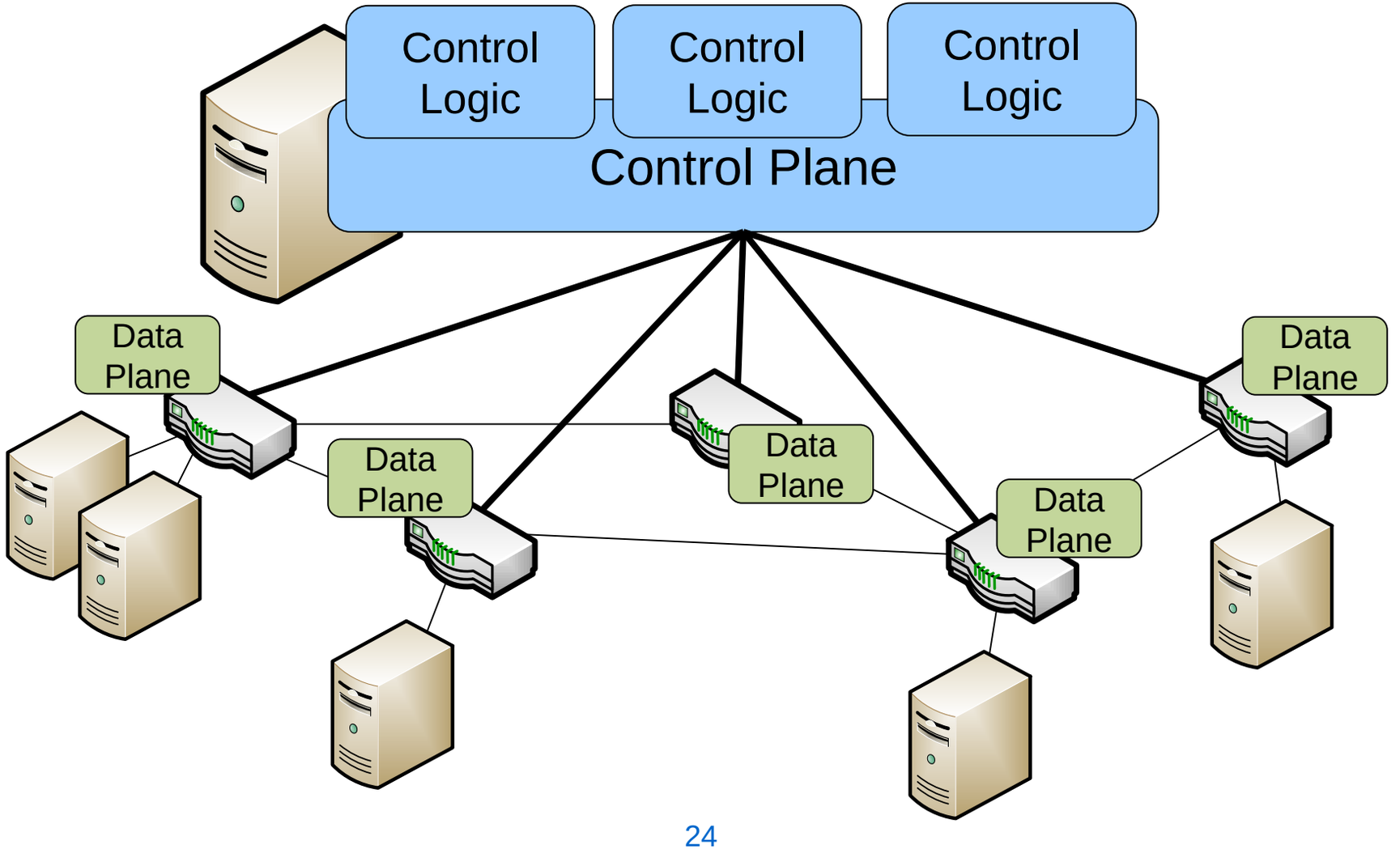
Forwarding defined on **flows** rather than only IP or MAC addresses

- Theoretically: any information of a packet that identifies a communication relation
- Practically: combinations of selected layer 2 to 4 header fields
  - Example: IP + Ports + protocol ids
- Fine-grained forwarding of selected flows or coarse-grained aggregation

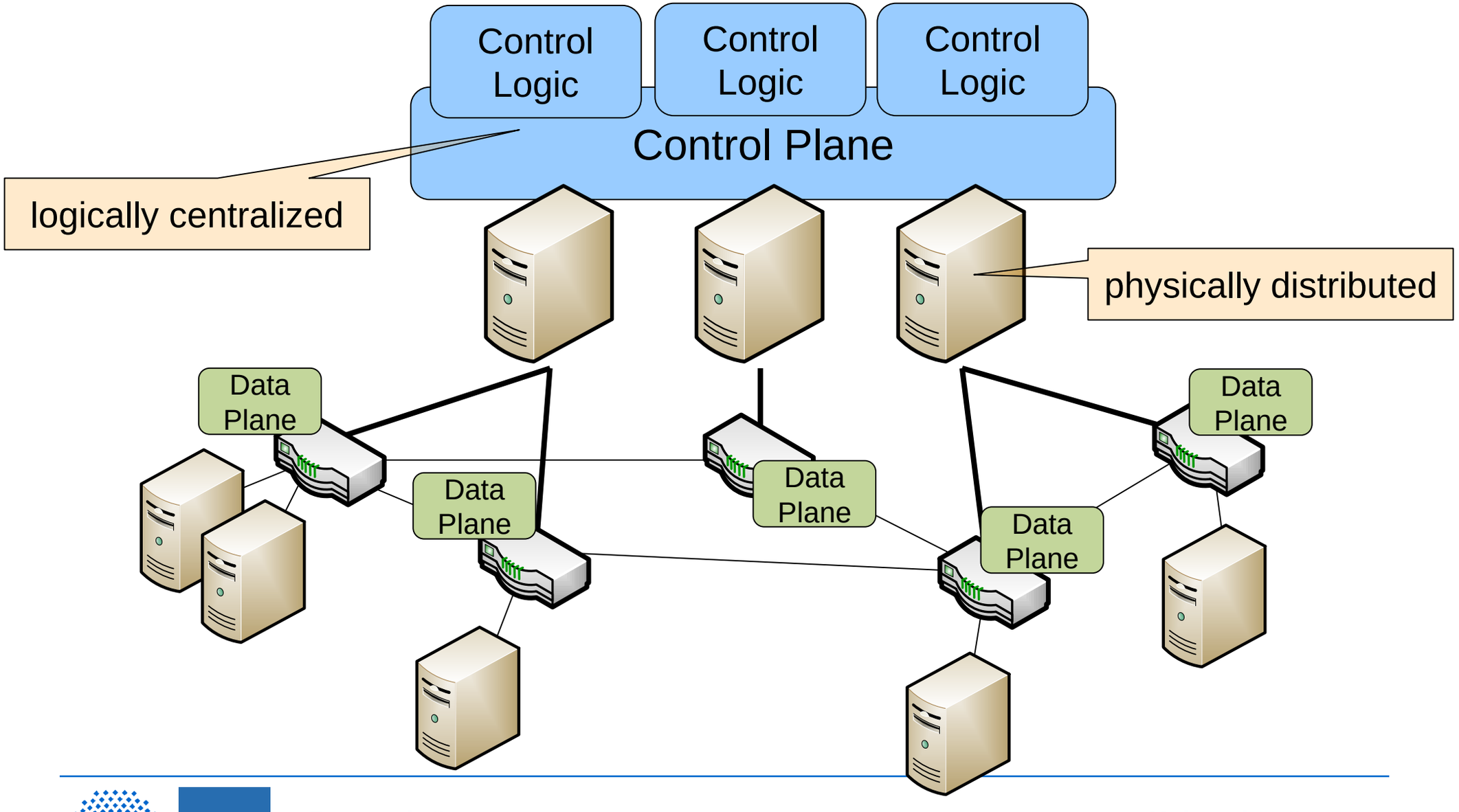


Control Plane and Data Plane Separation

- **Control plane:** defines routes, manages network graph
- **Data plane:** forwarding of packets



Logically Centralized Controller



Why Logical Centralization?

- Greatly simplifies implementation of control logic
  - Consistent global view onto network
- Global view can increase performance of control
  - E.g., faster convergence

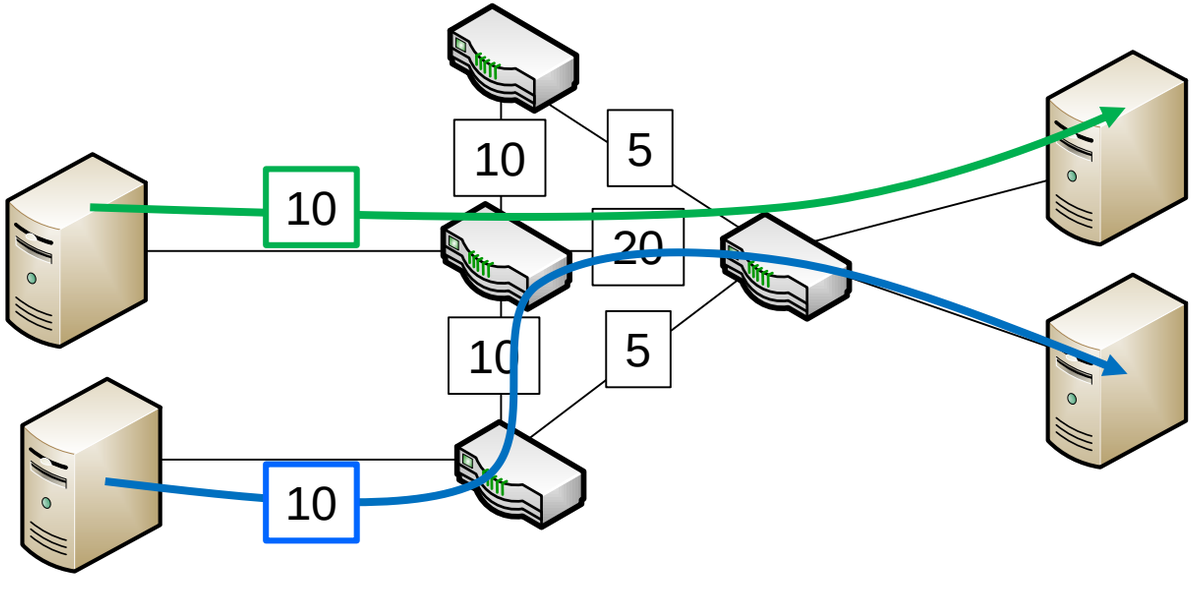
➔ Example: Distributed vs. centralized routing (see next slides)

- Physical distribution ensures high availability and scalability
  - Redundant controllers
  - Load distribution between controller instances

Distributed Routing (1)

Distributed routing protocols

- Need time to converge to optimum ➔ lower resource utilization
- Complex protocol and algorithm

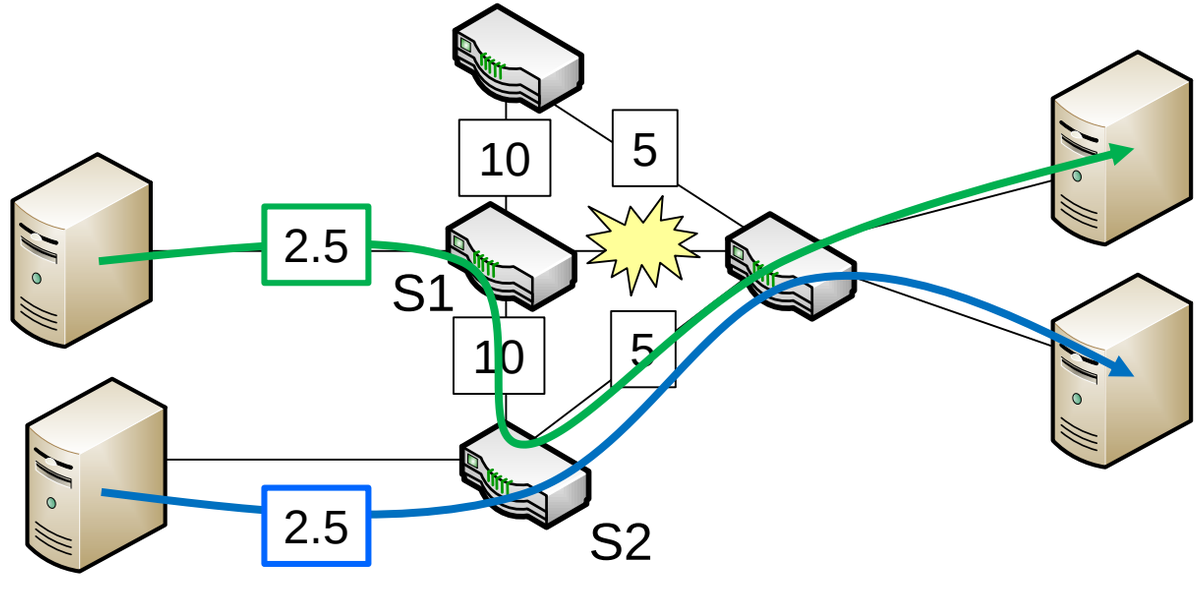


Distributed Routing (2)

Distributed routing protocols

- Need time to converge to optimum ➔ lower resource utilization
- Complex protocol and algorithm

Switches S1 & S2 decide to use the same paths

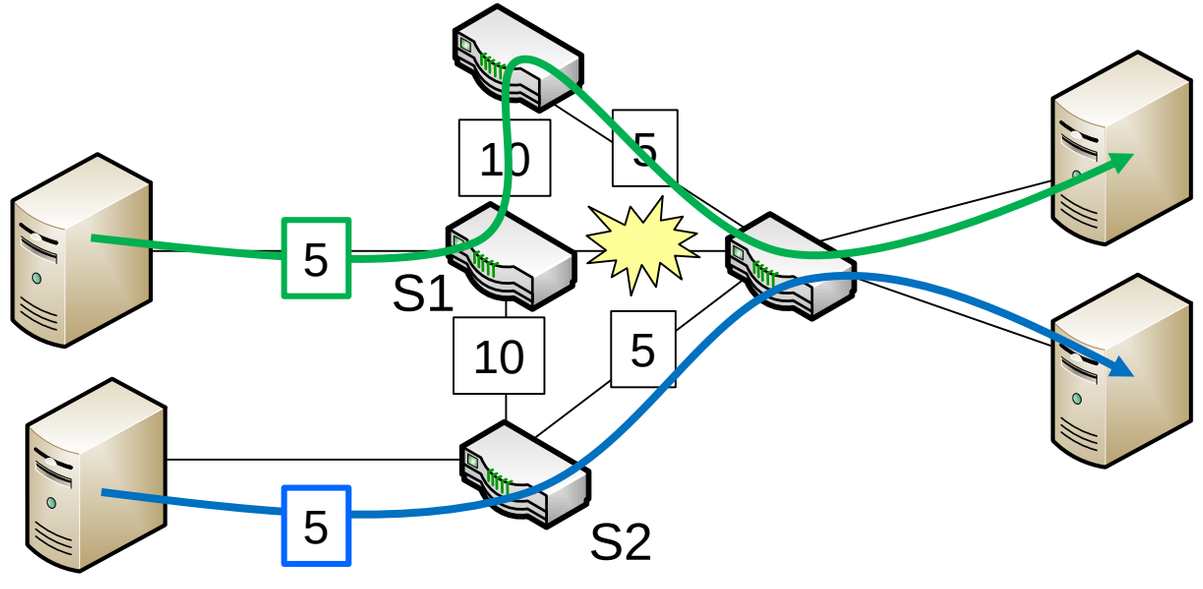


Distributed Routing (3)

Distributed routing protocols

- Need time to converge to optimum ➔ lower resource utilization
- Complex protocol and algorithm

Converged to optimum after 2 steps

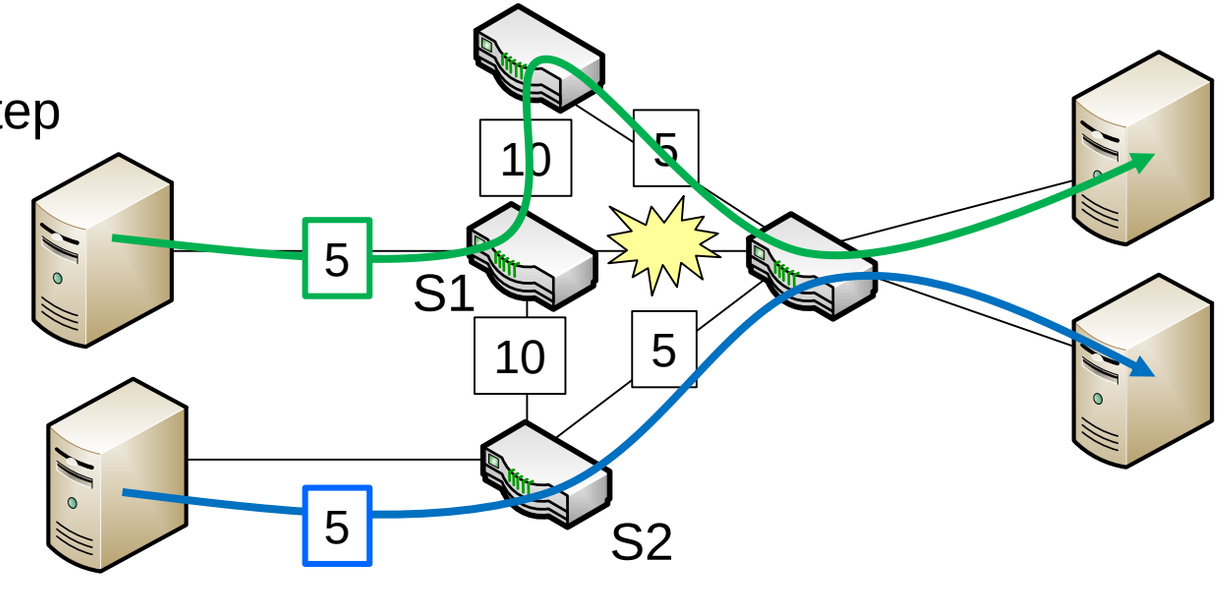


Logically Centralized Routing

Centralized optimization

- Faster convergence ➔ higher resource utilization
- Simpler routing algorithm on global view

Converged to optimum in 1 step



SDN in Practice

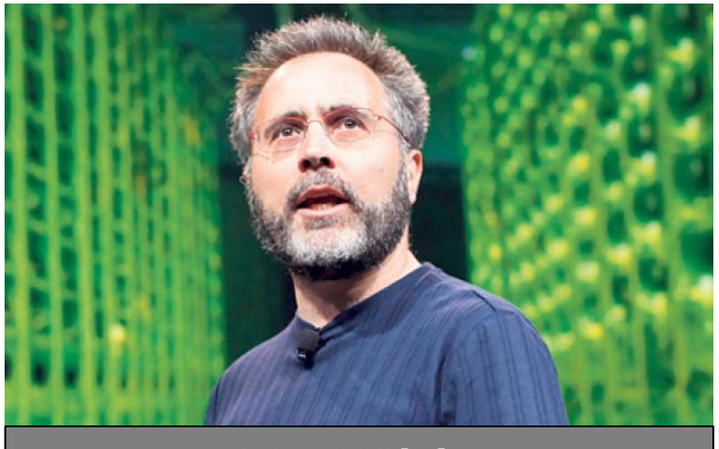
SDN used today to manage and optimize networks large networks

- Optimization of Google backbone
- Management of huge data center networks at Google, Microsoft, etc.

➔ Great improvement in manageability

➔ Significant performance increase

  - From < 40 % to nearly 100 % network utilization



Urs Hölzle
Senior Vice President for Technical Infrastructure @ **Google**

http://goo.gl/DN6RYv

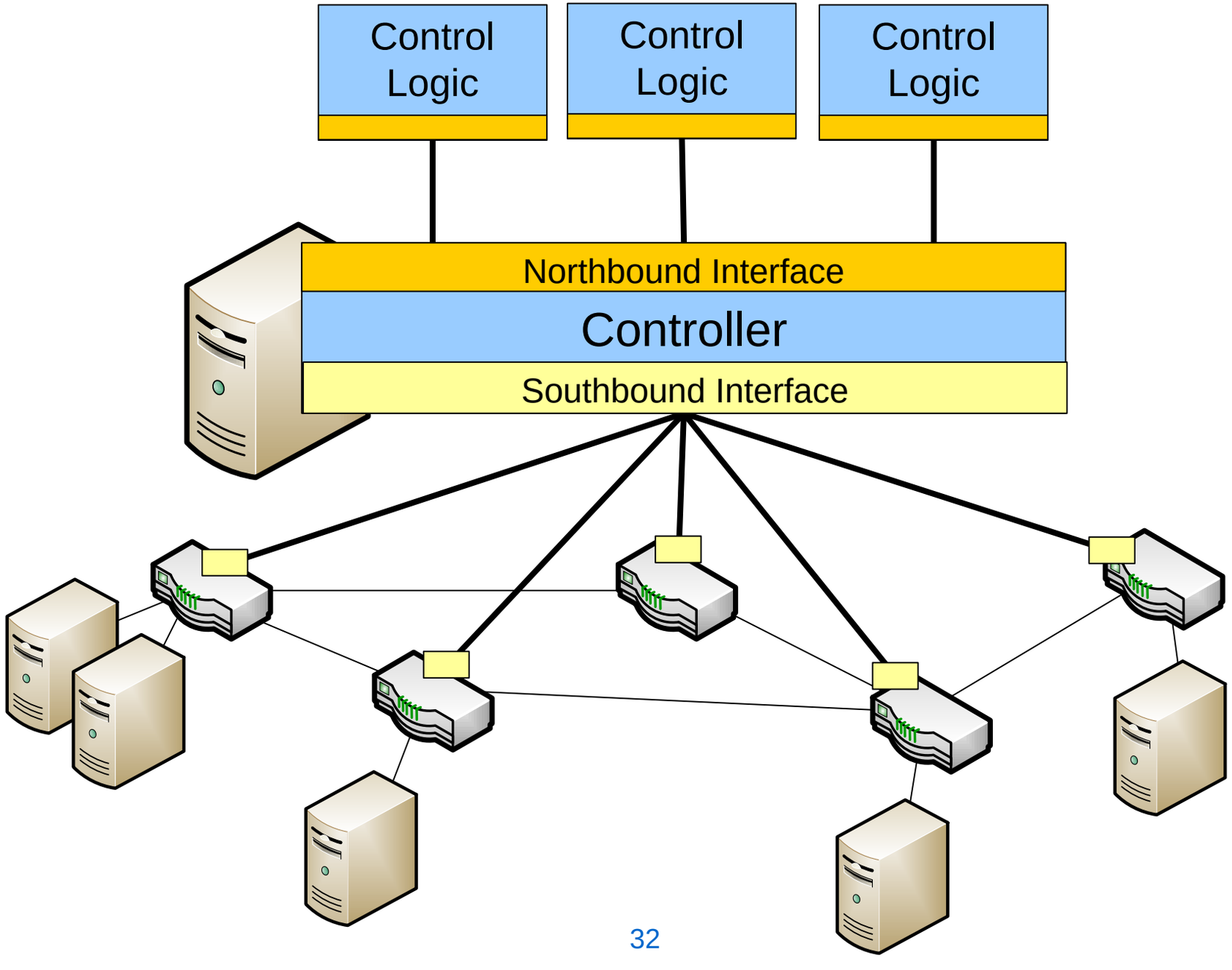
Amin Vahdat
Fellow and Technical Lead for Networking @ **Google**

https://goo.gl/3IqOGs

Mark Russinovich
CTO of Microsoft Azure

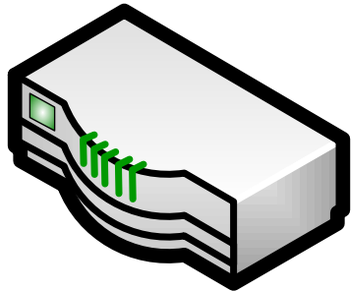
https://goo.gl/8fsule

Architecture of an SDN System



Components (1)

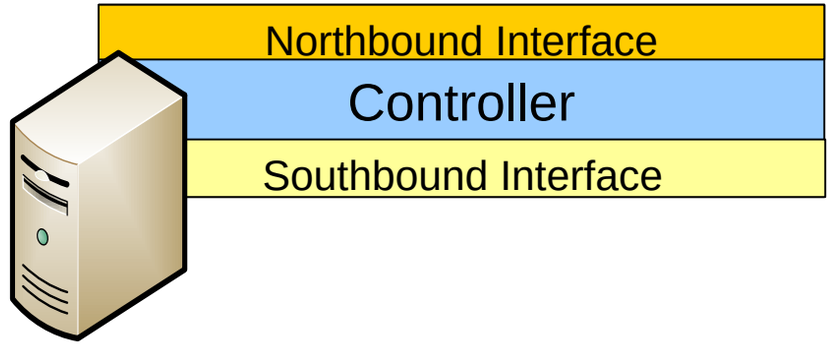
Switches/Routers



- Implement data plane: packet forwarding
  - Manages forwarding information base
- Typically multi-layer switches
  - Forwarding based on layer 2-4 headers
- Hardware switches
  - Hardware support for fast matching
    - Ternary content-addressable memory (TCAM)
- Software switches
  - Connect multiple virtual machines to physical interface of host
  - Example: Open vSwitch [http://openvswitch.org/]
- Hybrid switches: Implement SDN & standard L2/L3 forwarding

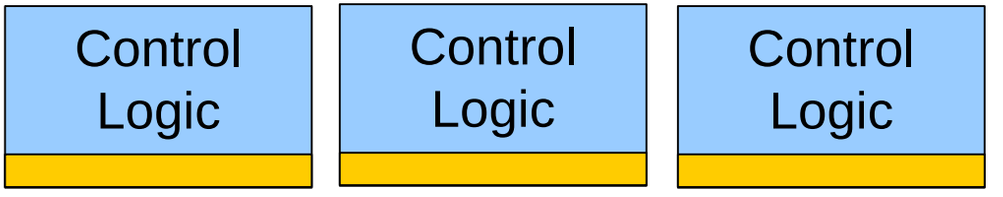
Components (2)

SDN Controller



- Server process executed on host
- Implements control plane
- Implements southbound interface to switches
  - Configuration of forwarding tables
  - Injecting packets
  - Events from switch (packet-in)
  - Collection of traffic statistics
  - Discover of toplogy
- Interfaces with control logic (control “application”) via northbound interface(s)

Components (3)



Control Logic

- Defines routes of “flows”
  - Proactive and reactive routing
- Might interface with other information sources
  - Example: Virtual machine manager knowing locations of VMs on hosts

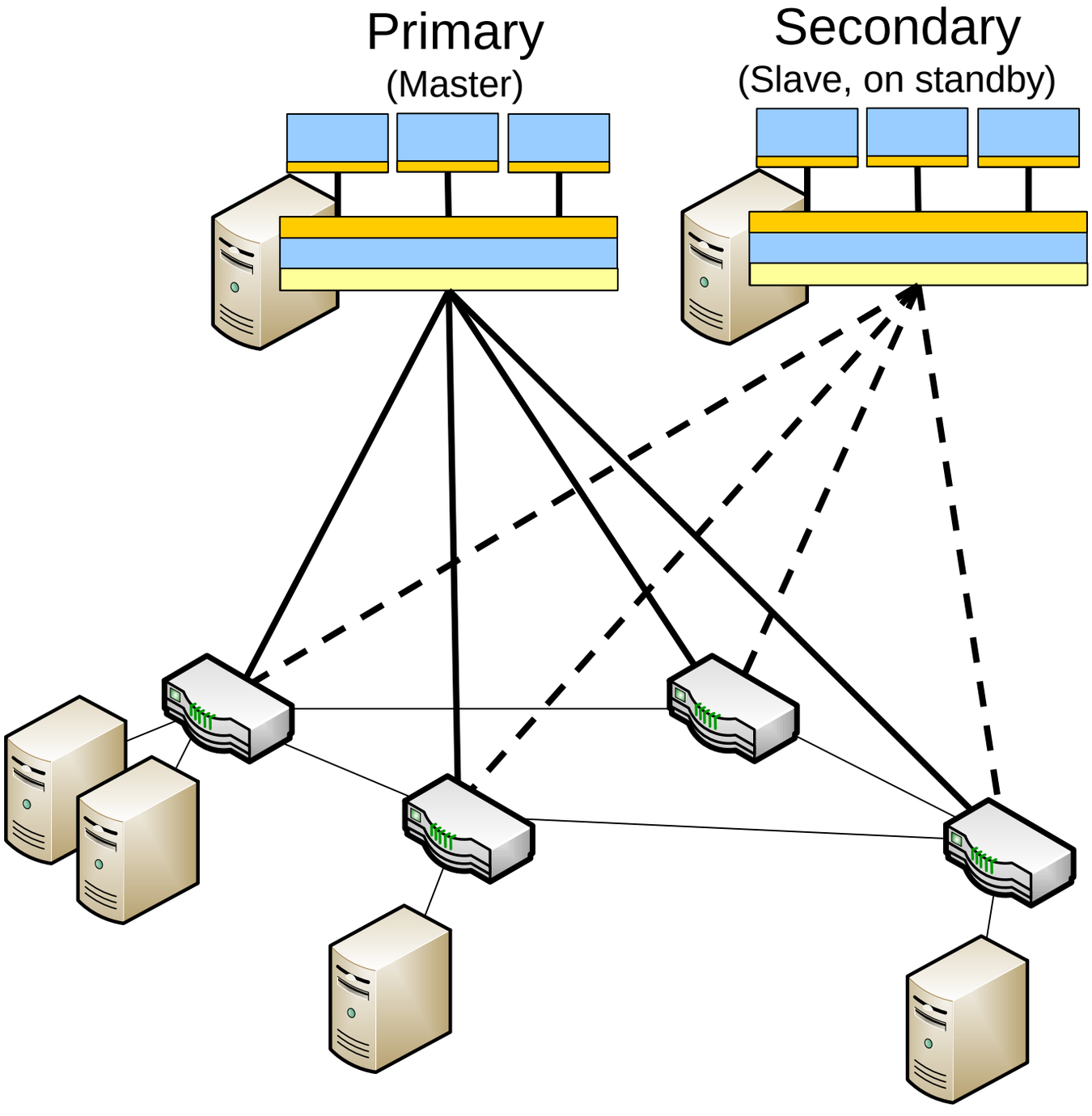
Control Plane Distribution

- Controller can be **physically distributed** in various ways
  - Improve robustness
  - Increase scalability

- Logical centralization ensures:
  Physical distribution is transparent to control logic
  - Allows for global view onto the system

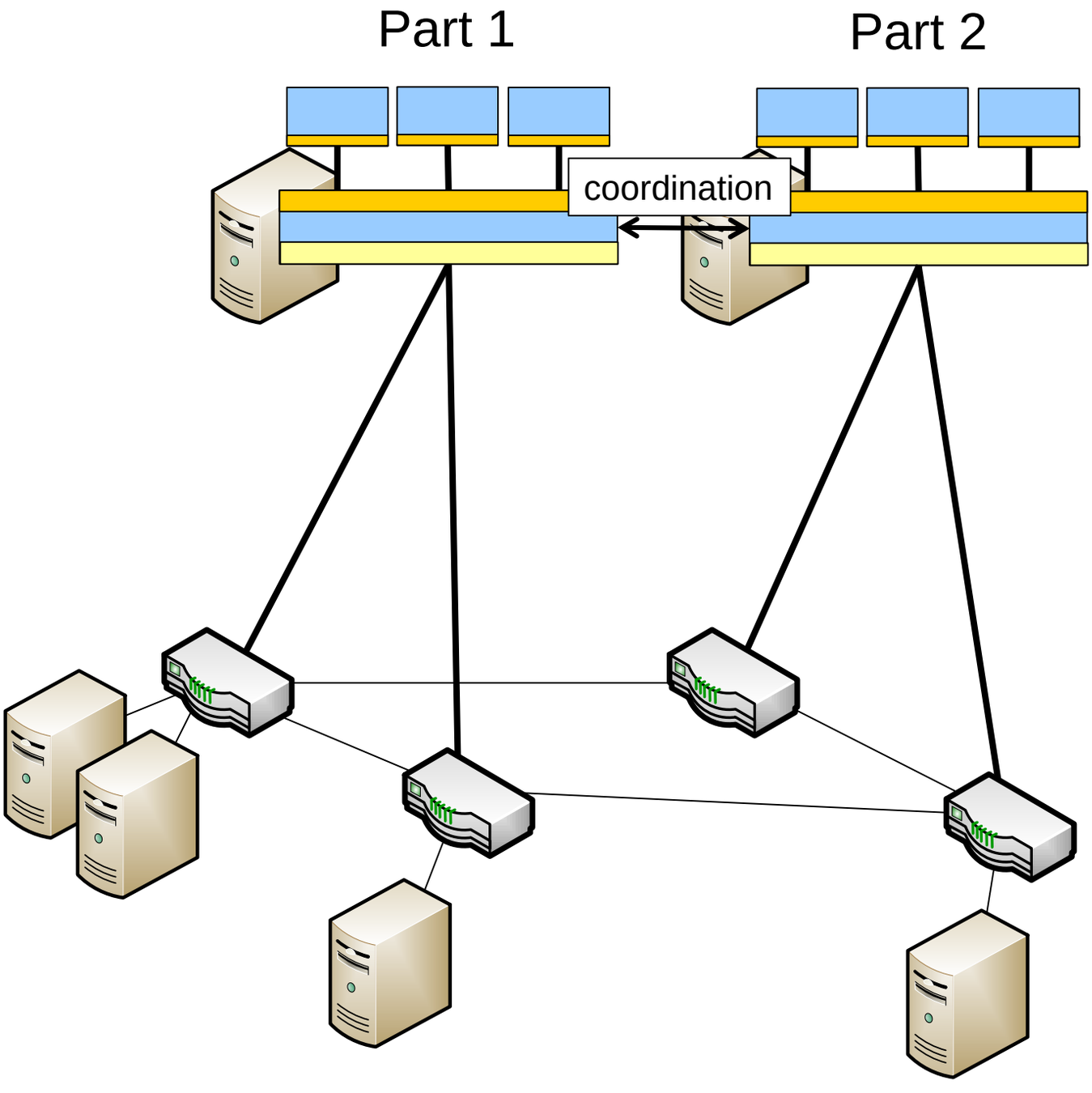
- No standard way of distribution defined

Control Plane Distribution: Replication



OpenFlow standard allows for connections from switch to multiple controllers

Control Plane Distribution: Partitioning



Support large-scale deployment

- Raises coordination problem
- How to achieve good performance (short paths) and consistent routes when paths span several parts?

Summary

Key concepts of SDN:

- Flow-based forwarding
  - Enables fine-grained treatment of flows
- Control/data plane separation
  - High flexibility without sacrificing performance
    - Simple adaptation of control logic implemented by controller software
    - Fast forwarding in hardware by switches
  - Simpler switches through “outsourcing“ of control logic
- Logically centralized control
  - Ease of control logic implementation through distribution transparency
  - Higher performance through global optimization

Questions?