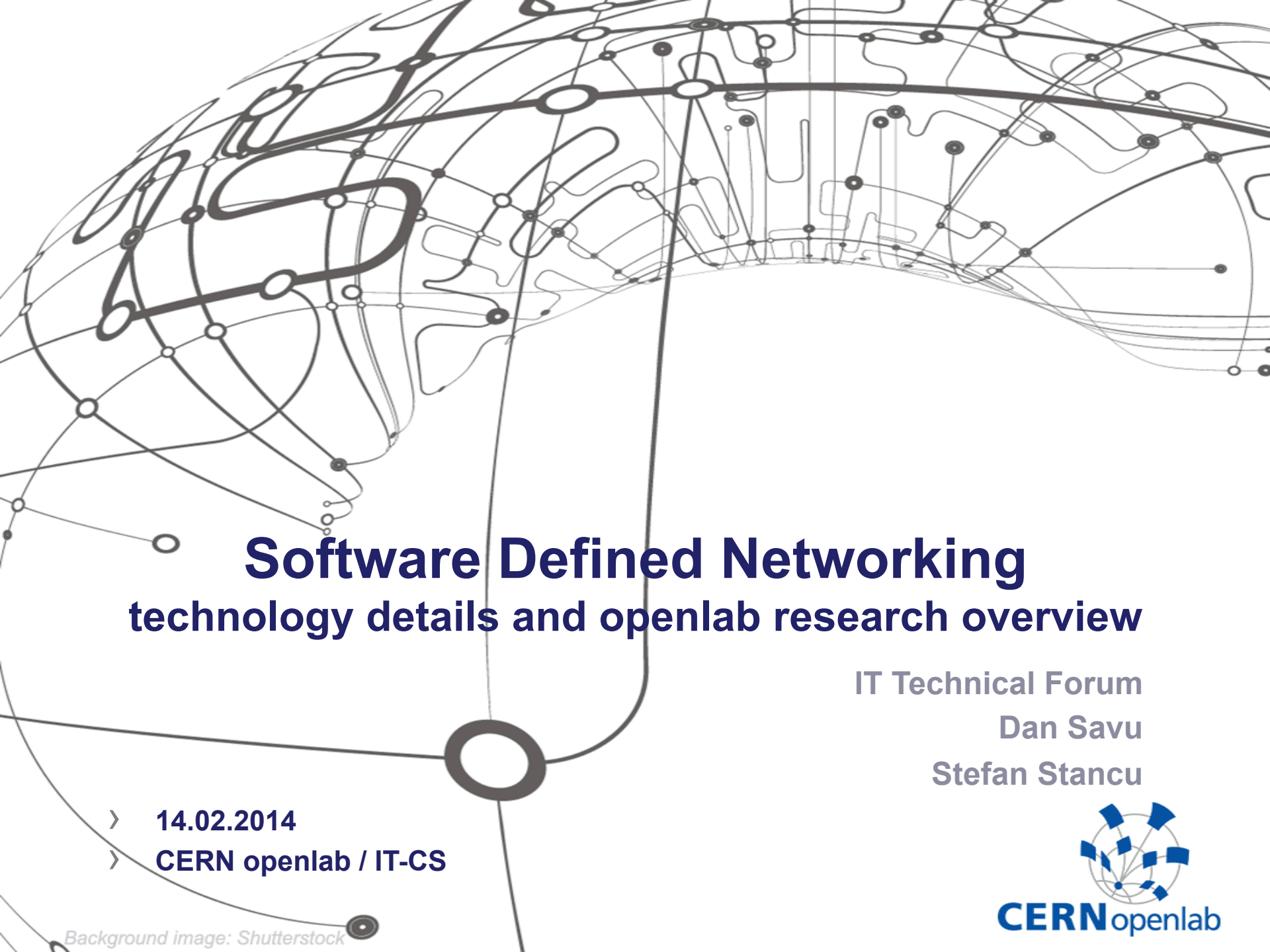


Software Defined Networking

technology details and openlab research overview

IT Technical Forum
Dan Savu
Stefan Stancu

- 14.02.2014
- CERN openlab / IT-CS

CERN openlab

Background image: Shutterstock

Outline

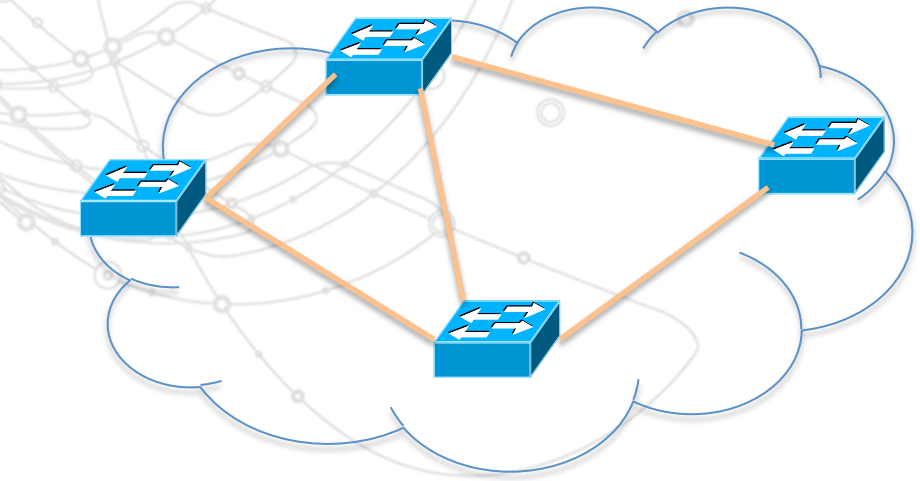
- **Software Defined Networking**
  - From traditional networking to SDN
  - The OpenFlow protocol
  - SDN Hardware and Software
  - SDN vs. NFV vs. NV
  - IP Routing Services and Openflow (RouteFlow)
- **The openlab ViSION project**

Data plane

Traditional

Packet streaming

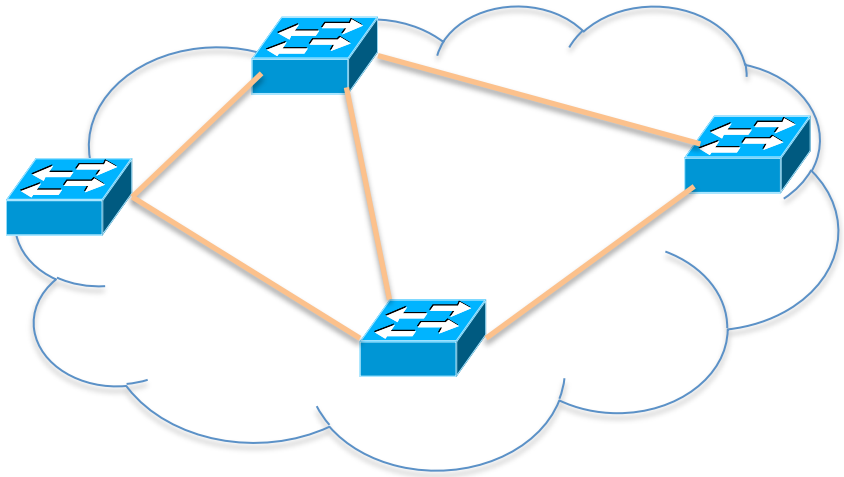
- Forward, based on tables matching
- Tables are “closed” in the devices
- Filter, buffer, rate-limit, measure



SDN

Similar but

- Tables are “open”
- Table format and actions clearly specified
- Well defined API



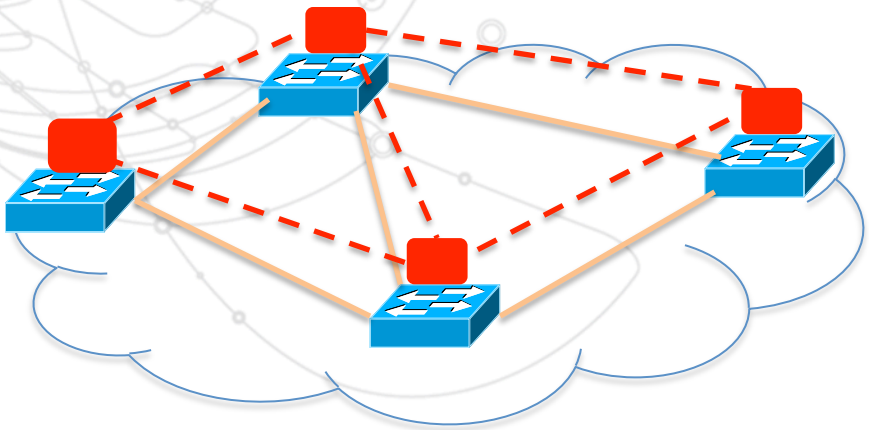
Control Plane

Traditional

Fully distributed protocols

Each device must:

- Track topology changes
- Compute routes
- Install forwarding rules (tables contents)



SDN

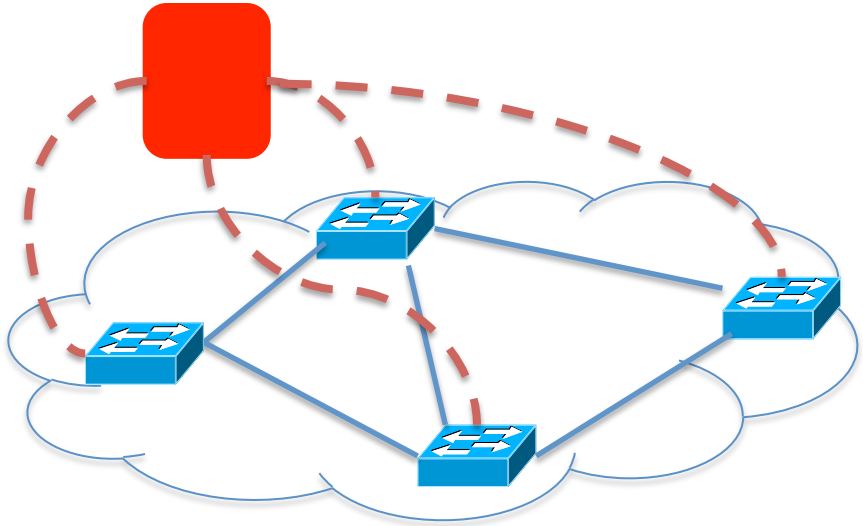
Logically-centralized control

Software controller

Open API to Data Plane

- e.g. OpenFlow

“Dumb” fast switches



Management Plane

Traditional

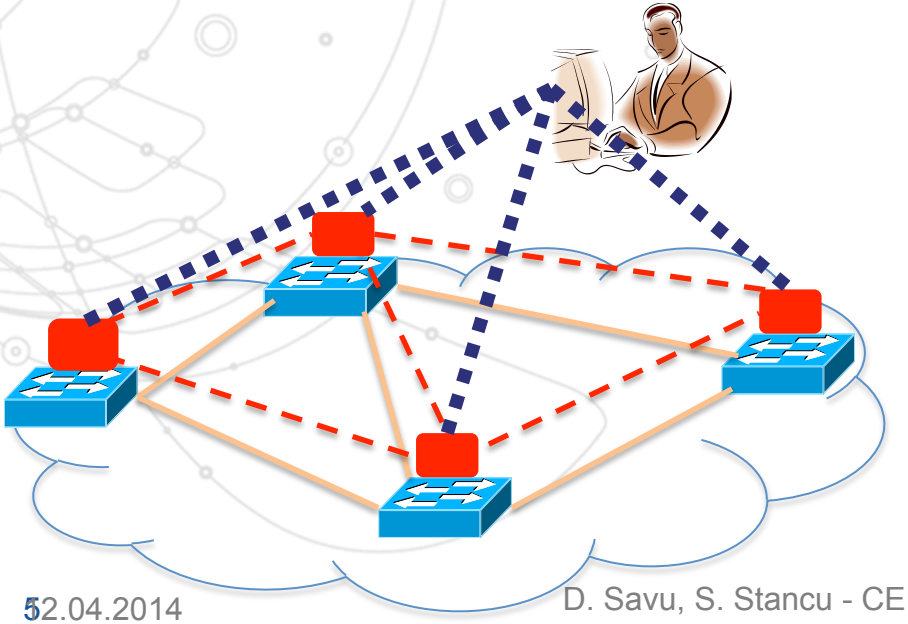
Configuration

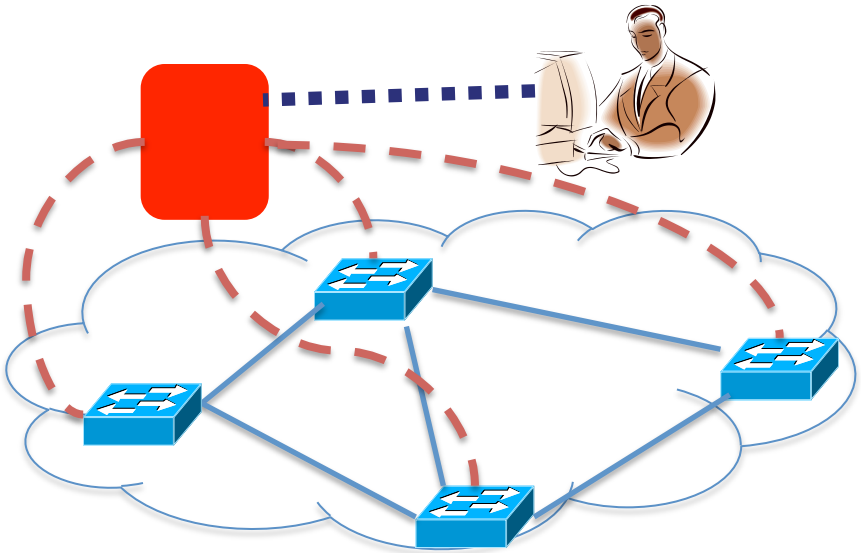
- Proprietary interfaces
- Device by device configuration
- Automation possible but tedious
  - CLI parsing

SDN

Configuration

- Central control
- Single interface (API) to all devices
- One stop shop





Business cases

- **If my Hadoop cluster is running, then:**
  - allocate a set of dedicated paths through the network
  - move all other traffic to other links or services provided it doesn't reduce bellow an SLA.

- **Every 24 hours:**
  - transfer 10 TB of data to the backup service
  - avoid impact on any other services

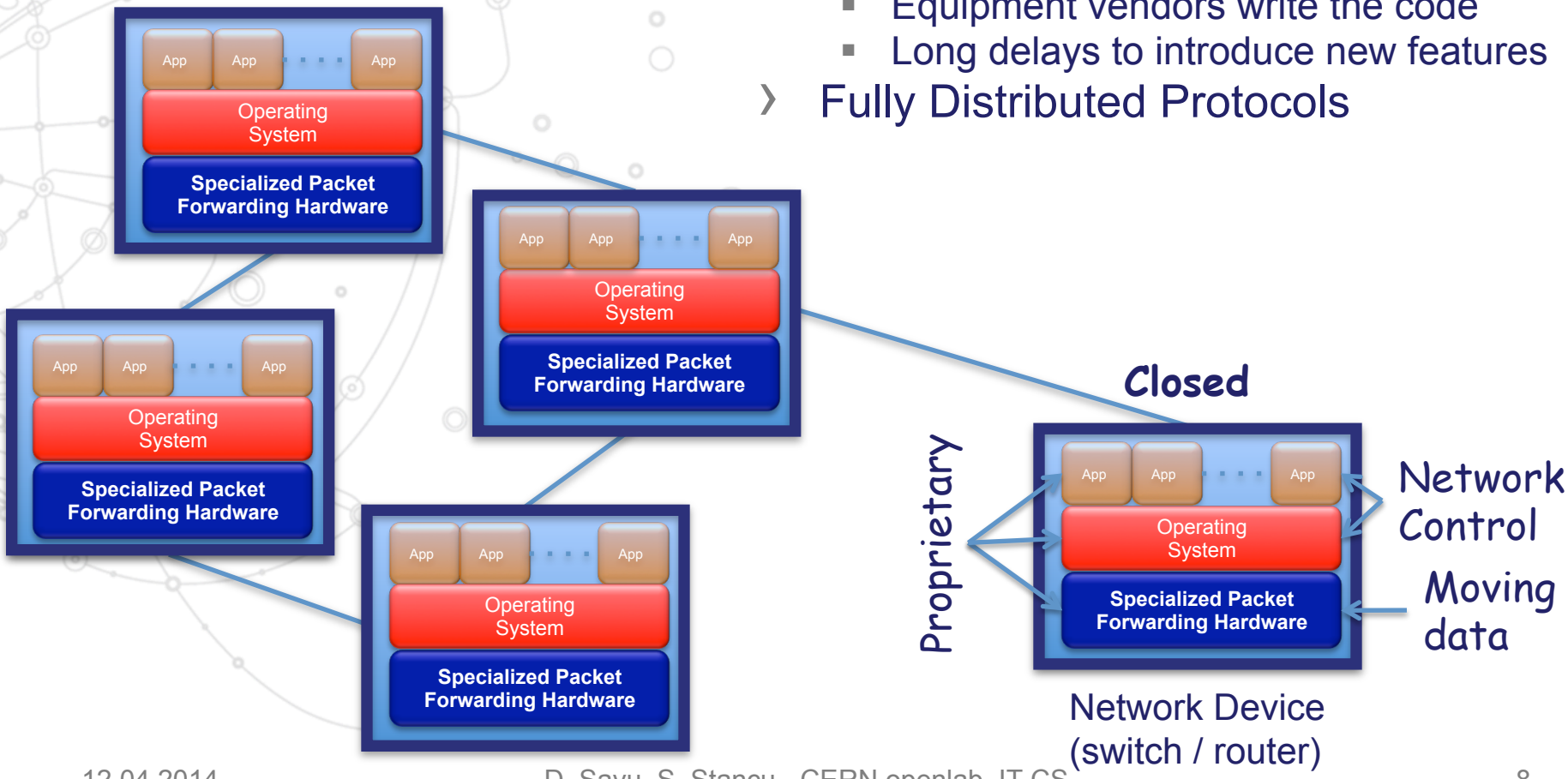
- **I have a new prototype application or hardware:**
  - create a new isolated network
  - must not interfere with normal network activity

Outline

- Software Defined Networking
  - From traditional networking to SDN
  - The OpenFlow protocol
  - SDN Hardware and Software
  - SDN vs. NFV vs. NV
  - IP Routing Services and Openflow (RouteFlow)
- The openlab ViSION project

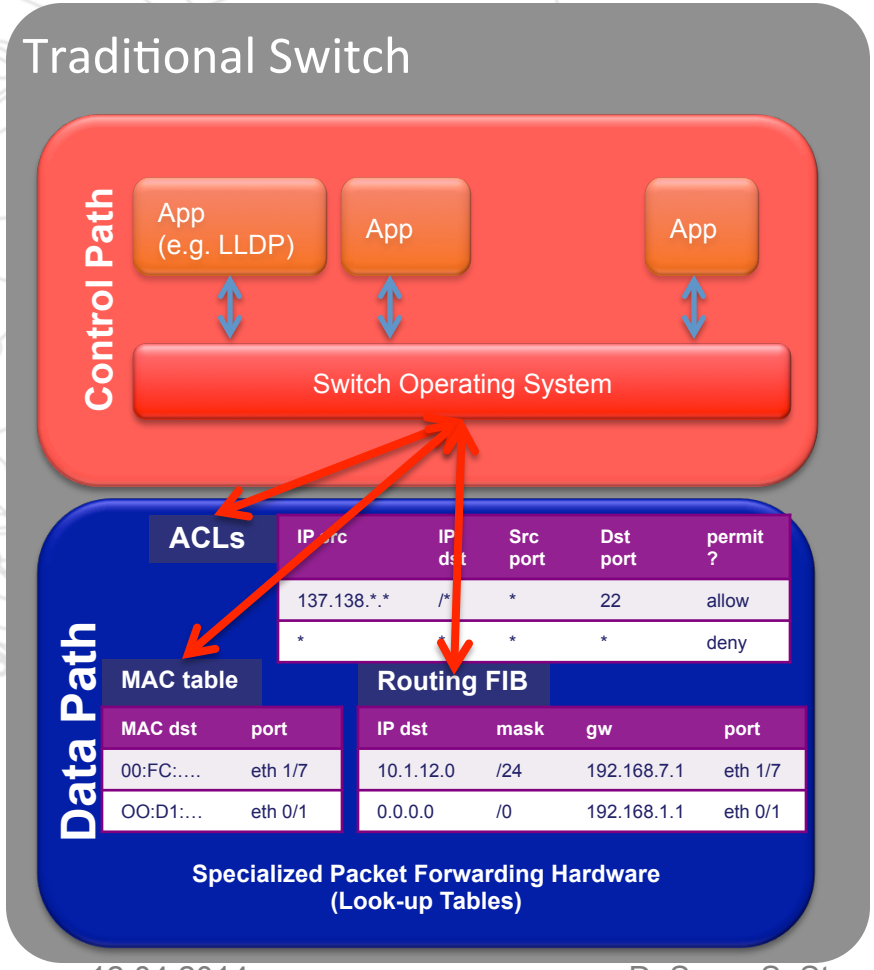
Networking Nowadays

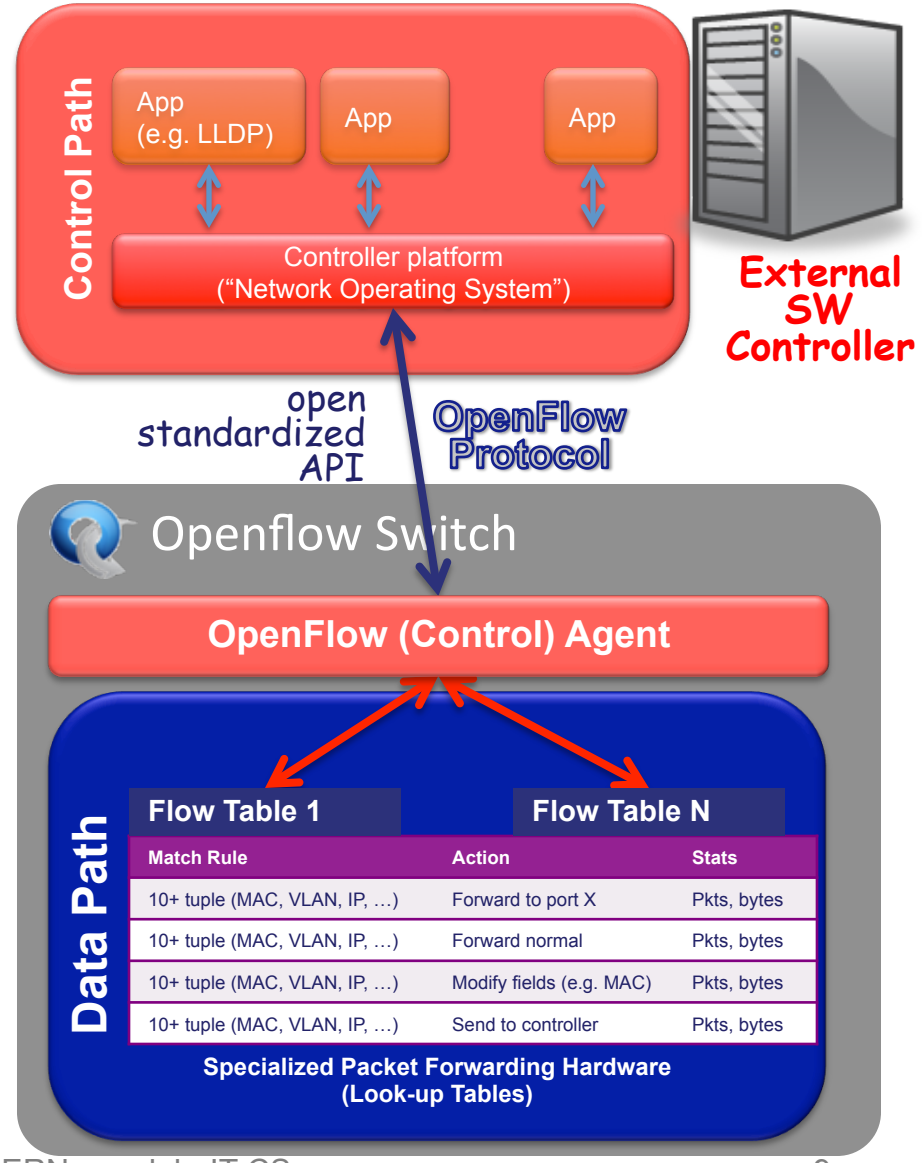
- Closed equipment
  - Software bundled with hardware
  - Vendor-specific interfaces
- Few people can innovate
  - Equipment vendors write the code
  - Long delays to introduce new features
- Fully Distributed Protocols



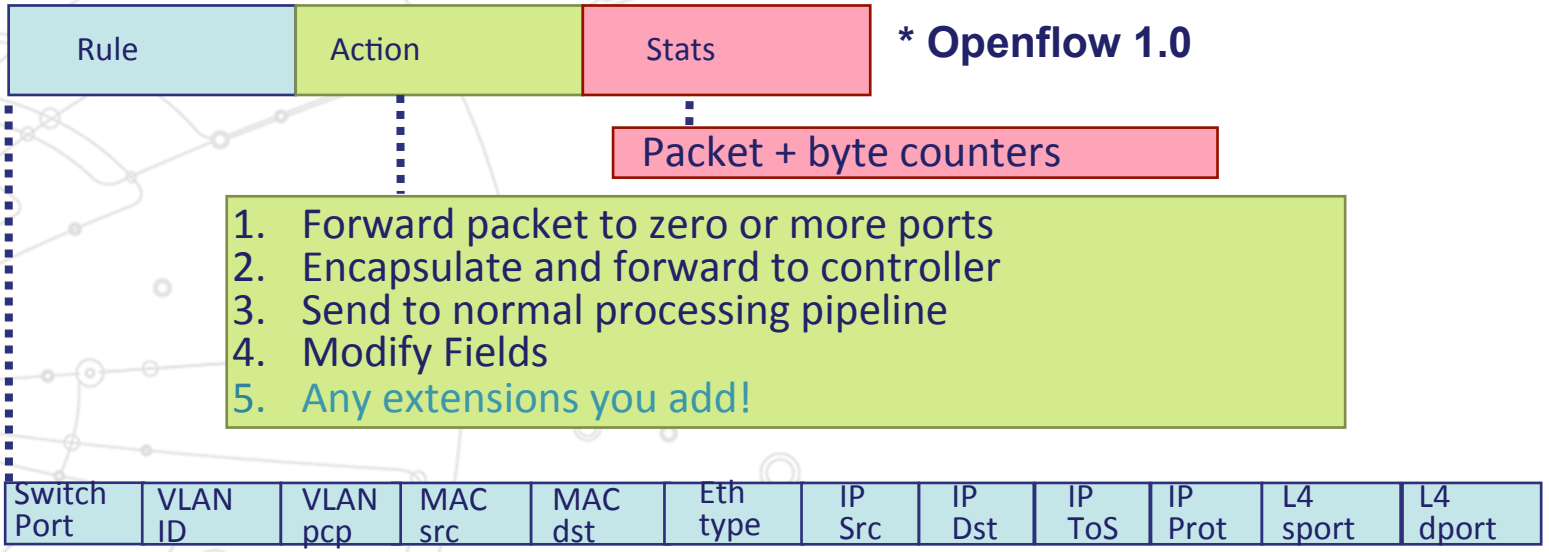
Background image: Shutterstock

OpenFlow – Decouple Control & Data



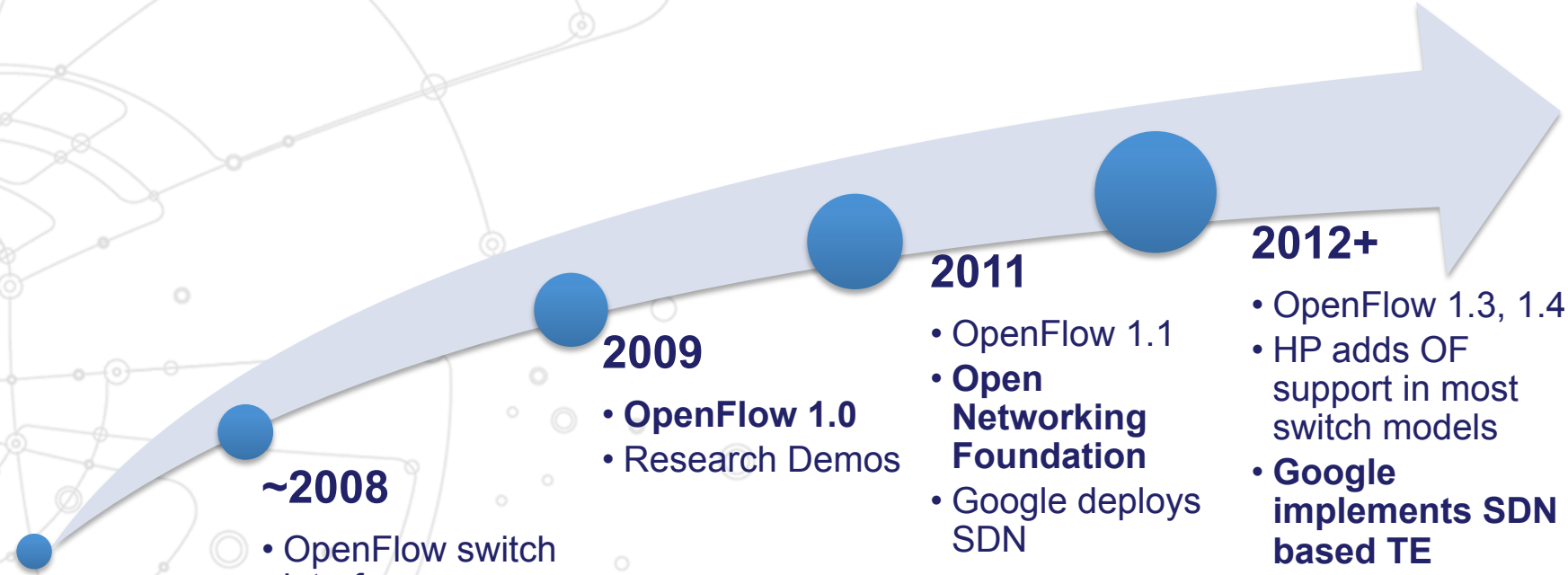


OpenFlow* – Flow Table Entries



MAC SRC	MAC DST	IP SRC	IP DST	TCP SRC PORT	TCP DST PORT		Actions	Count
*	00:08:	*	*	*	*	…	Port 1	250
*	*	*	10.2.2.	*	*	…	Port 5	350
*	*	192.168.	*	*	23	…	Drop	678
*	*	10.1.	192.168	*	*	…	Rewrite Header	654
*	FF:…:FF	*	*	*	*	…	Flood	56
*	*	*	*	*	*	…	Controller	11

SDN & Openflow Evolution



~2004

- Early work on new management paradigms (e.g. Ethane, SANE)

~2008

- OpenFlow switch interface
- NOX Operating System

2009

- **OpenFlow 1.0**
- Research Demos

2011

- OpenFlow 1.1
- **Open Networking Foundation**
- Google deploys SDN

2012+

- OpenFlow 1.3, 1.4
- HP adds OF support in most switch models
- **Google implements SDN based TE**

Born on the campus, maturing in the data center !

ONF Members

The Open Networking Foundation is dedicated to the promotion and adoption of Software-Defined Networking (SDN) through open standards development.

Background image: Shutterstock

OpenFlow Evolution

- **ONF (Open Networking Foundation) is the body maintaining the OpenFlow specs.**

- **12/2009**
  - **Widely deployed**
- L1-L4 field matching
- Actions: send to port, rewrite L2-L4 headers, set/strip VLAN tag
- Subnet masks
- Vendor Extensions
- Controller failover
- Barrier command
- Multiple queues per output port
- TLS

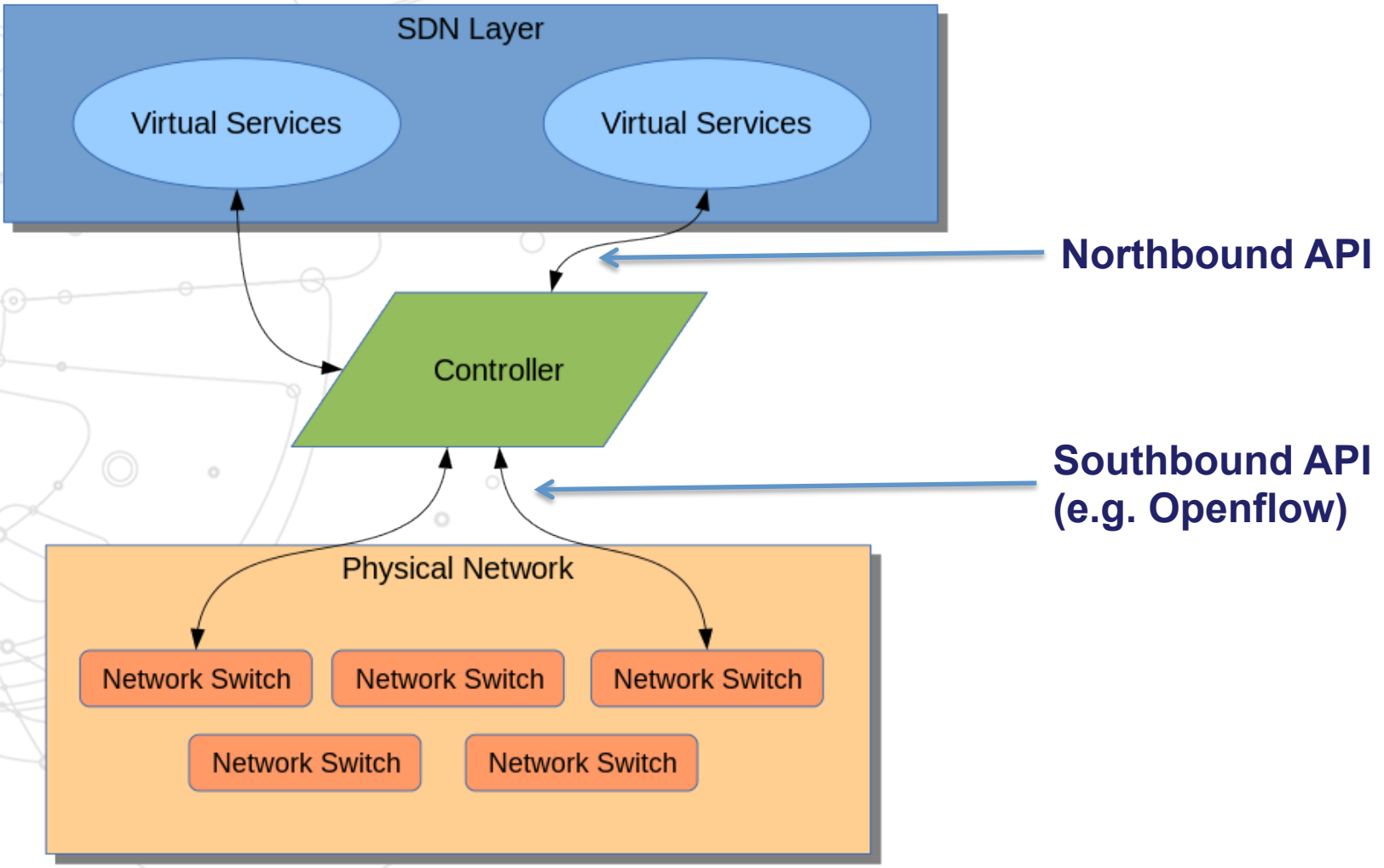
- **02/2011**
- Multiple tables
- Multi-level VLAN and MPLS tagging
- Virtual ports
- Port groups
- Controller connection failure

- **12/2011**
- Extensible match support (TLV)
- Basic IPv6 support
- Message bundles
- Controller role change mechanism

- **04/2012**
  - **Next gen ASICs**
- IPv6 extension header
- Per-flow meters
- Eviction and vacancy events
- MPLS BoS match
- Tunnel-id metadata
- Packet-in cookies

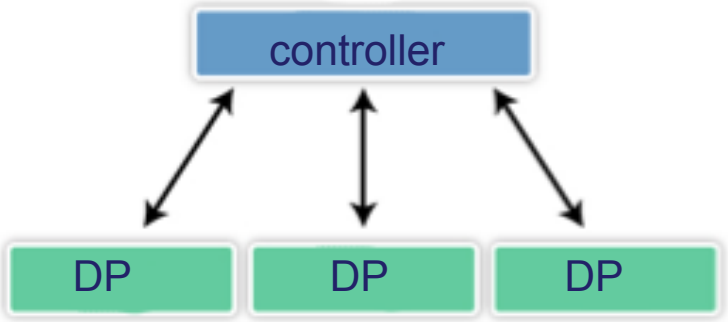
- **08/2013**
- Optical port properties
- IANA TCP port 6653
- Controller flow monitoring and status events

SDN Overview

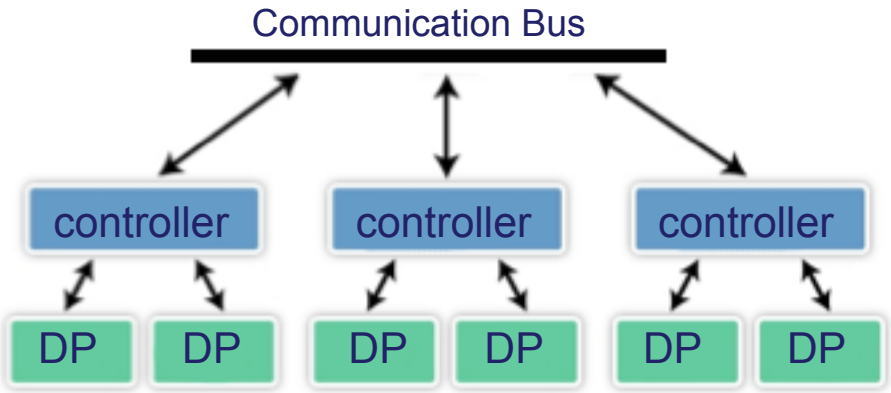


SDN Controller Topologies

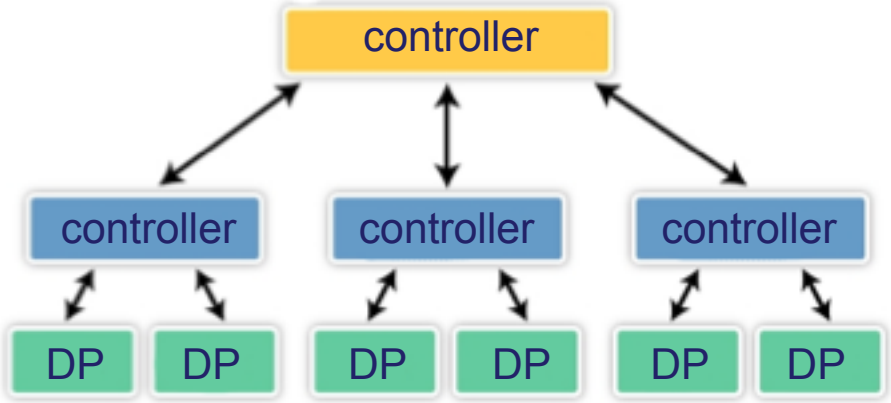
CENTRALIZED



DISTRIBUTED



HIERARCHICAL



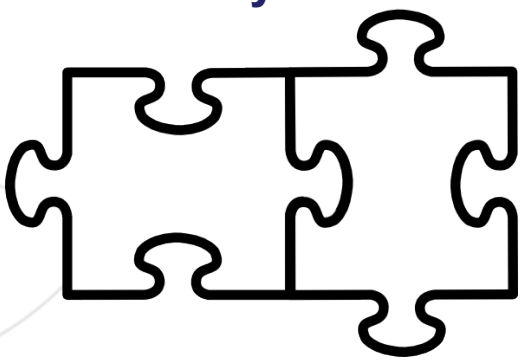
DP = Data Path
(openflow switch)

Reactive vs. Proactive

- First frame triggers controller for flow entry creation
- Small flow table
- Flow setup latent
- Controller availability vital

- Controller generates flow table for architecture
- Lower latency
- Less flexible/ Dynamic
- Wildcard flow entries

Hybrid



Outline

- Software Defined Networking
  - From traditional networking to SDN
  - The OpenFlow protocol
  - SDN Hardware and Software
  - SDN vs. NFV vs. NV
  - IP Routing Services and Openflow (RouteFlow)
- The openlab ViSION project

Openflow Switches *

Company	Switch
HP	5400zl, 8200zl, 6200yl, 3500 3500yl, 6600
Brocade	MLXe, CER, CES
Ciena	Coredirector w/ firmware 6.1.1
Cisco	Cat6k, Catlyst 3750, 6500 series
Juniper	MX, EX, T-640
Arista	EOS, 7050, 7124FX
NEC	IP8800, PF5240, PF5820
Pronto	3240, 3290, 3295, 3780
Toroki	Lightswitch 4810
Dell	Z9000, S4810
Quanta	LB4G
Extreme summit	X440, x460, x670
Huawei	Openflow capable platform
IBM	8264
NetGear	7328SO, 7352SO

*** most switches have some of the openflow features implemented in software (forwarding capacity drastically reduced)**

SDN Open Source Controllers

Functionally Oriented (little or no support for high availability, scaling, etc)		
	Language	**Examples**
	C/C++	**NOX**, Trema (also Ruby) and MUL
	Java	**Beacon**, Maestro and **Floodlight**
	Ocaml	Mirage and Frenetic
	Haskell	Nettle, McNettle and NetCore
	Python	**POX**, RYU and Pyretic
	JavaScript	**NodeFlow** (for Node.JS)

Enterprise Grade		
	Controller	**Details**
	OpenDaylight (Linux Foundation project)	Joint industry effort. Virtually all the big players are contributing members members Release v 1.0 (02/2014)
	ON.LAB ONOS (Open Networking Operating System)	Floodlight based Work in progress High availability, distributed, scale-out

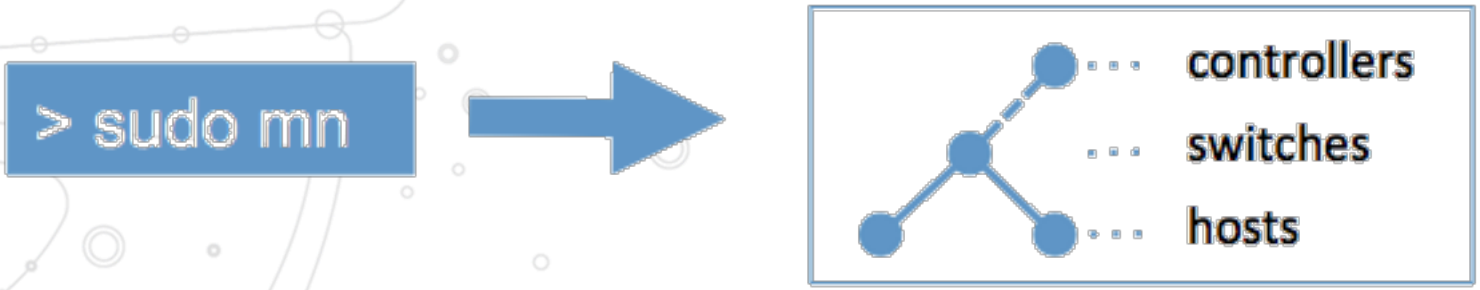
SDN Commercial Controllers

Company	SDN Controller
HP	VAN (Virtual Application Networks) OpenFlow 1.3 support High Availability Infrastructure controller SDN ecosystem
Big Switch Networks	Big Network Controller
Cisco Systems	XNC (Extensible Network Controller)
IBM	Programmable Network Controller
NEC	ProgrammableFlow Controller
NTT	Data Virtual Network Controller
Netsocket	vFlow Controller
Nicira (VMware)	NVP (Network Virtualization Platform)
Nuage Networks	VSC (Virtualized Services Controller)
Plexxi Inc	Plexxi Control
Pluribus Networks	Netvisor
Türk Telekom Group	YakamOS

* List from sdn central directory

Do Try This at Home

- **Mininet**
  - http://mininet.org/
  - Realistic virtual network environment
  - Real kernel, switch and application code on a single VM



- **Open vSwitch**
  - http://vswitch.org/
  - Production quality virtual switch, **OpenFlow**
  - Multi-server virtualized environment, development and testing
  - Part of Linux kernel as of 3.3
    - default switch in Xen Cloud Platform
    - integrated in OpenStack

Outline

- Software Defined Networking
  - From traditional networking to SDN
  - The OpenFlow protocol
  - SDN Hardware and Software
  - SDN vs. NFV vs. NV
  - IP Routing Services and Openflow (RouteFlow)
- The openlab ViSION project

SDN vs. NFV vs. NV

SDN – Software Defined Networking

- Separates control and forwarding planes, centralized view
- Routing, switching, management and provisioning the network
- Currently openflow is the main southbound protocol
- The ASIC in the switch plays an important part

NV – Network Virtualization

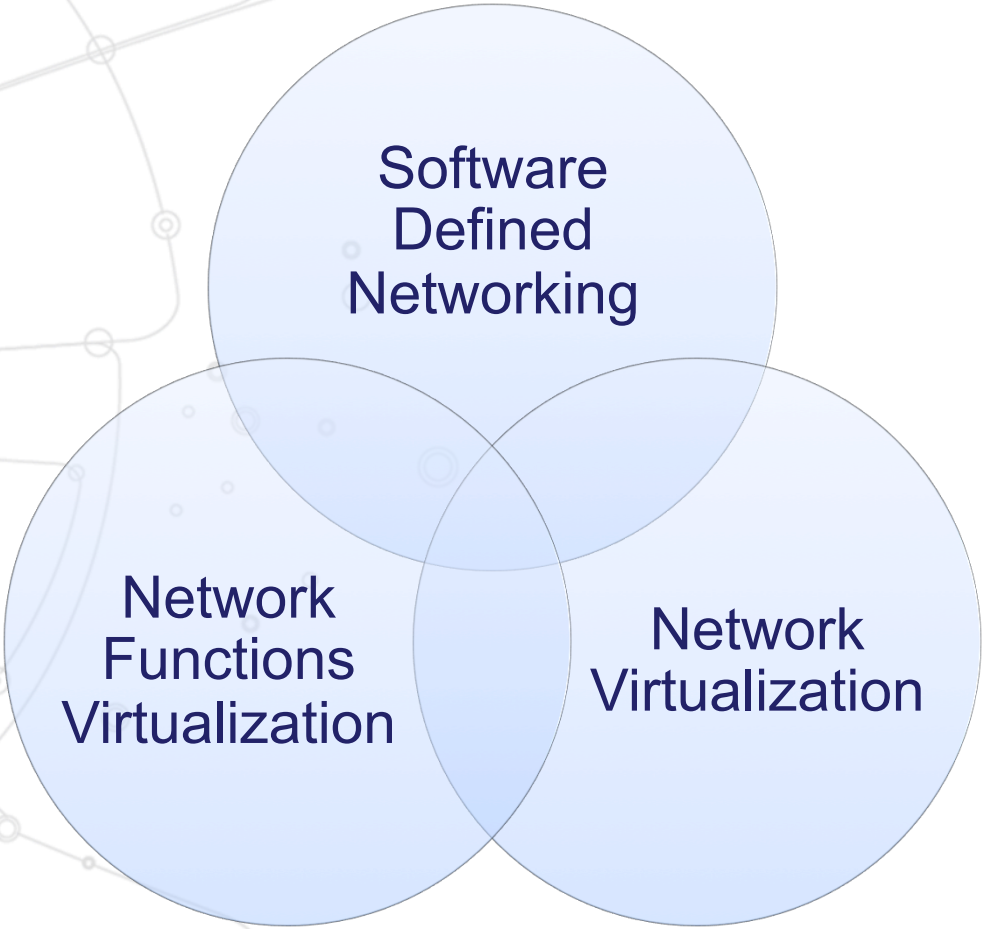
- Focused on building tunnels (aka overlays) to support virtualized architectures, multi-tenant sites etc.

NFV – Network Functions Virtualization

- Building virtual network functions into the overlays (e.g. DNS, firewall, gateway, SLA assurance etc.)

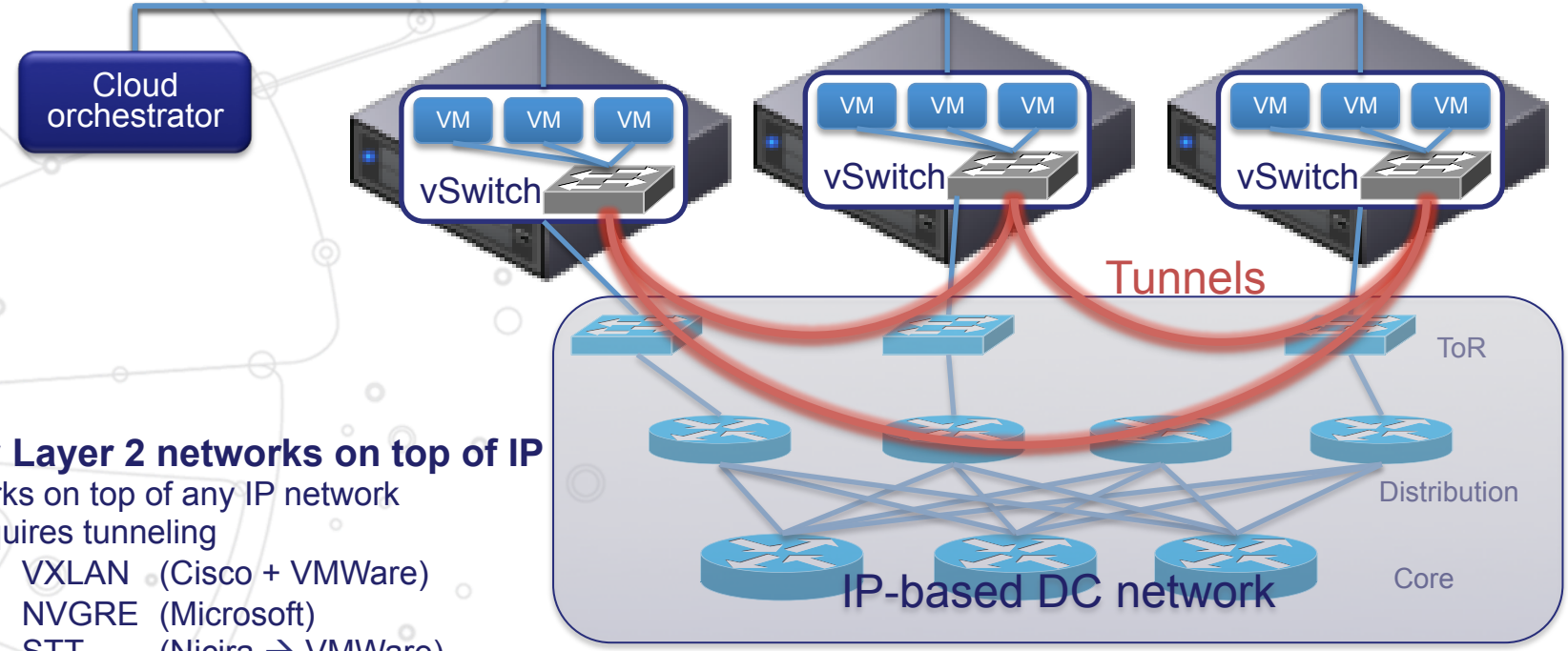
The goal is Network Agility !

SDN vs. NFV vs. NV



Reduces CAPEX, OPEX, Space and power consumption.
Competitive supply of innovative applications.

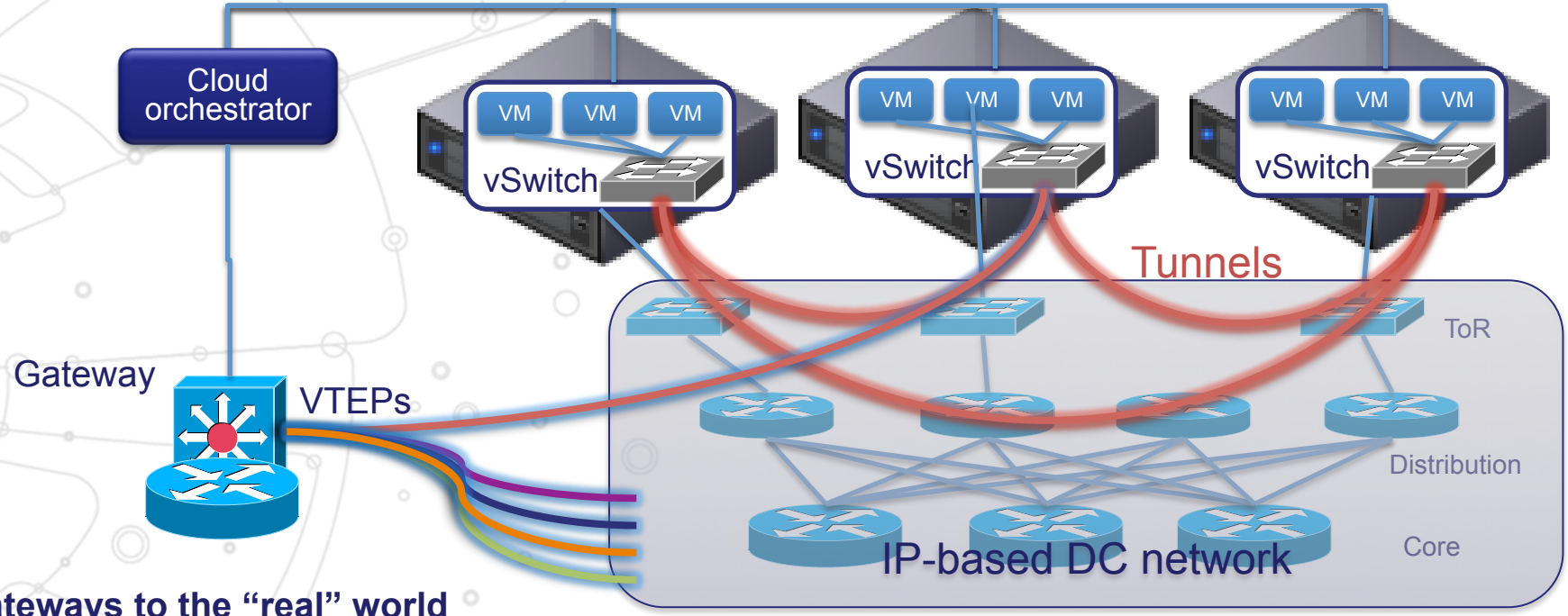
Network Virtualization



- **Overlay Layer 2 networks on top of IP**
  - Works on top of any IP network
  - Requires tunneling
    - VXLAN (Cisco + VMWare)
    - NVGRE (Microsoft)
    - STT (Nicira → VMWare)
  - Encapsulation done in Hypervisor switches
    - VTEP (Virtual Tunnel End Point)

- **Q&A**
  - Can I have a single overlaid L2 network over my entire DC?
    - NO: max 250-500 hosts per layer 2 domain
  - Can I move my VMs anywhere?
    - YES: as long as the Hypervisor vSwitch supports the chosen tunnel technology
  - How do I talk to the non-virtual world (storage, Internet, etc.)?
    - Good question: through gateways

Network Virtualization – Gateways



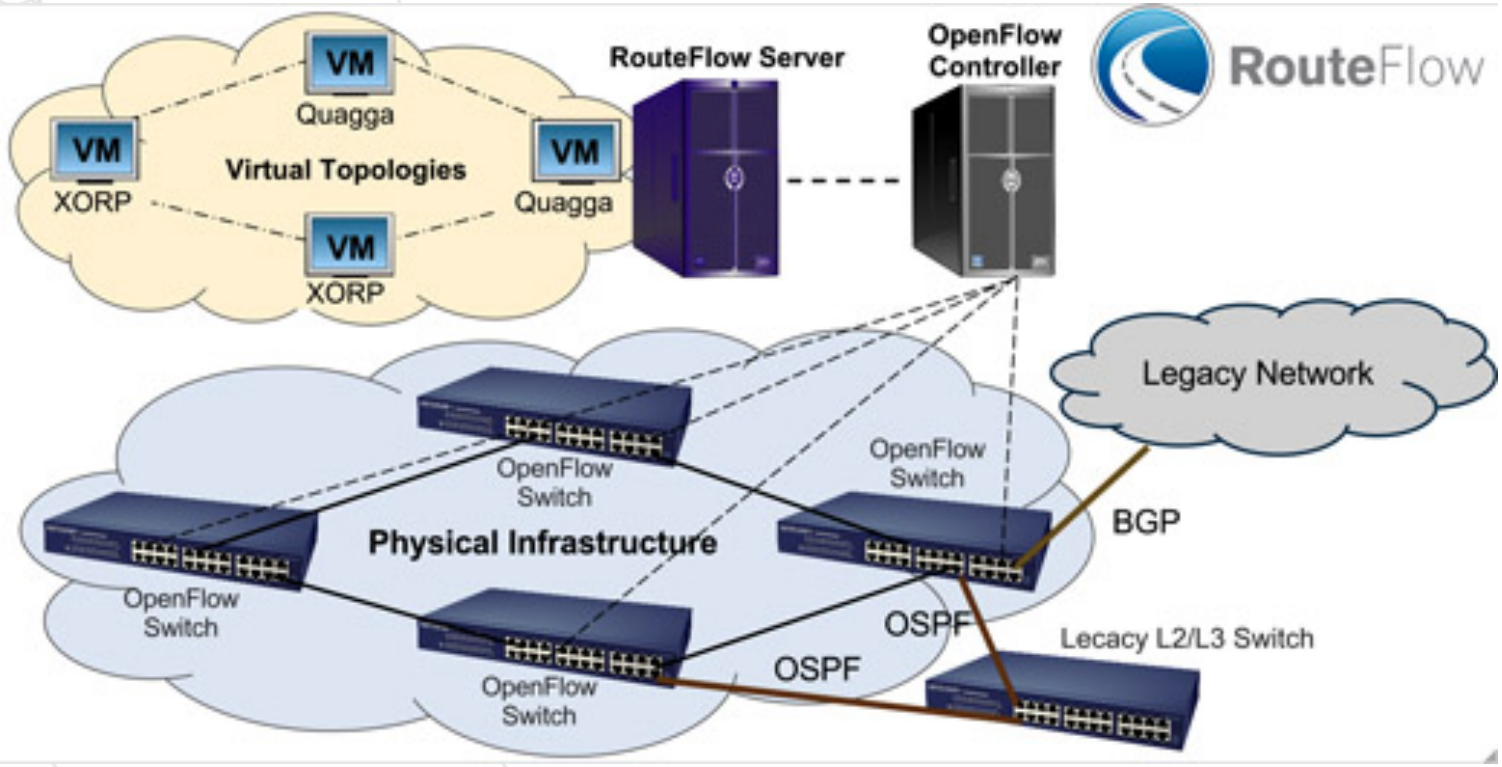
- **Gateways to the "real" world**
  - Required for
    - Connecting to the internet
    - Adding non virtualized appliances to the DC environment
  - Status:
    - Software gateways exist for some time (NICIRA)
    - Switches supporting VXLAN are coming to market
  - By definition any gateway is a potential bottleneck:
    - Throughput
    - Scaling with the number of VTEPs

Outline

- Software Defined Networking
  - From traditional networking to SDN
  - The OpenFlow protocol
  - SDN Hardware and Software
  - SDN vs. NFV vs. NV
  - IP Routing Services and Openflow (RouteFlow)
- The openlab ViSION project

Routeflow

Open source project to provide virtualized IP routing services over OpenFlow enabled hardware.

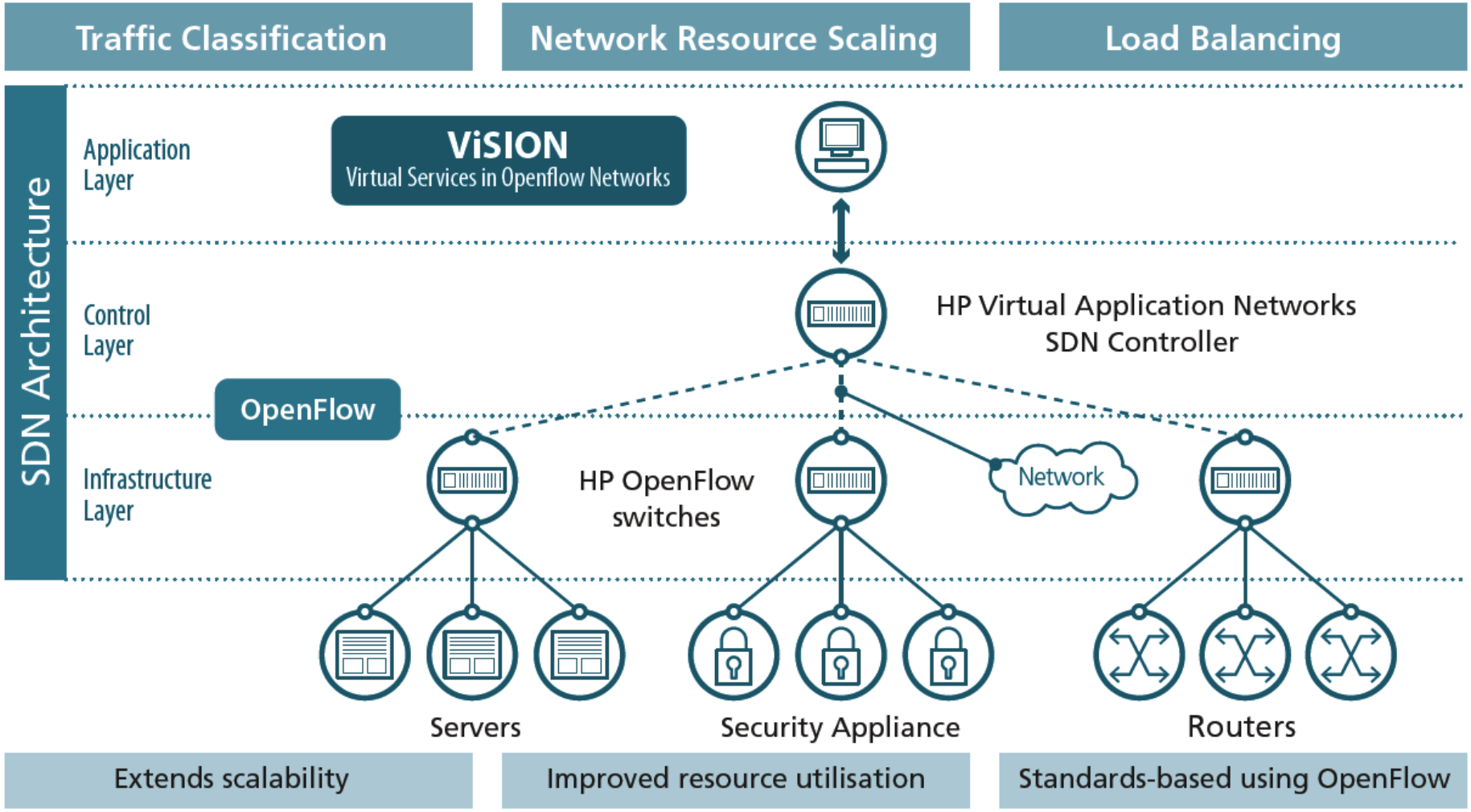


Outline

- Software Defined Networking
  - From traditional networking to SDN
  - The OpenFlow protocol
  - SDN Hardware and Software
  - SDN vs. NFV vs. NV
  - IP Routing Services and Openflow (RouteFlow)

- **The openlab ViSION project**

ViSION - HP SDN Framework



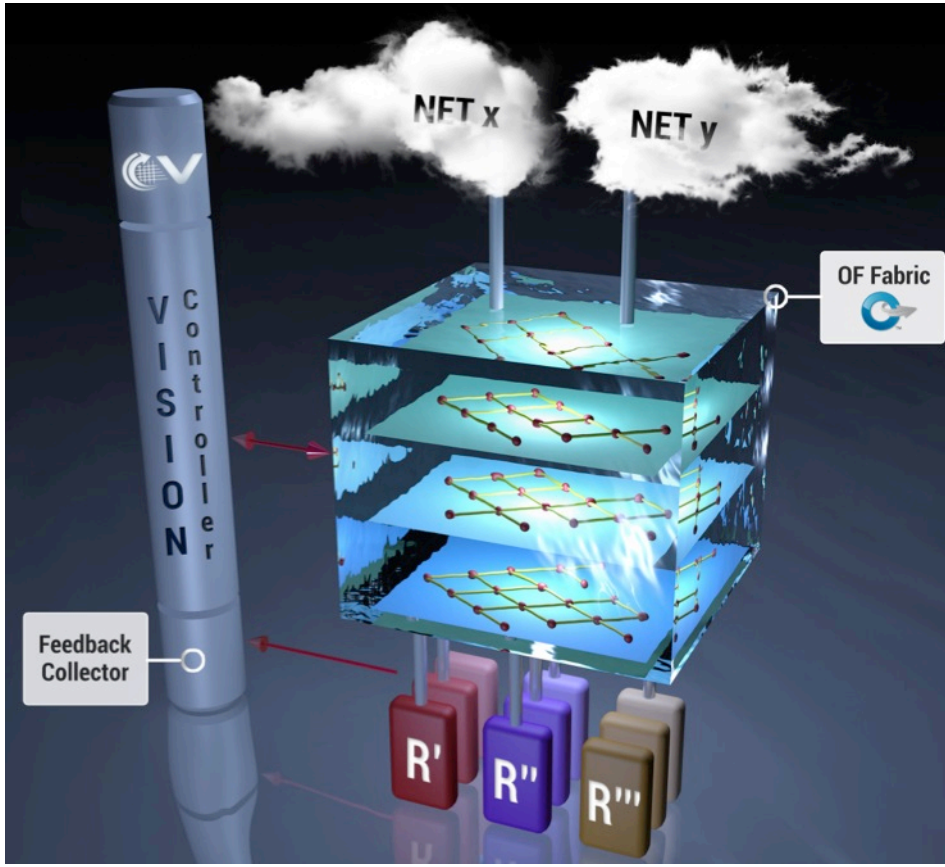
Background image: Shutterstock

ViSION Traffic Orchestrator

- **Traffic orchestration**
  - OF fabrics interconnect:
    - Client Networks
    - Resource pools
  - Vision Controller:
    - “programs” flows through fabrics
    - collects feed-back from resources

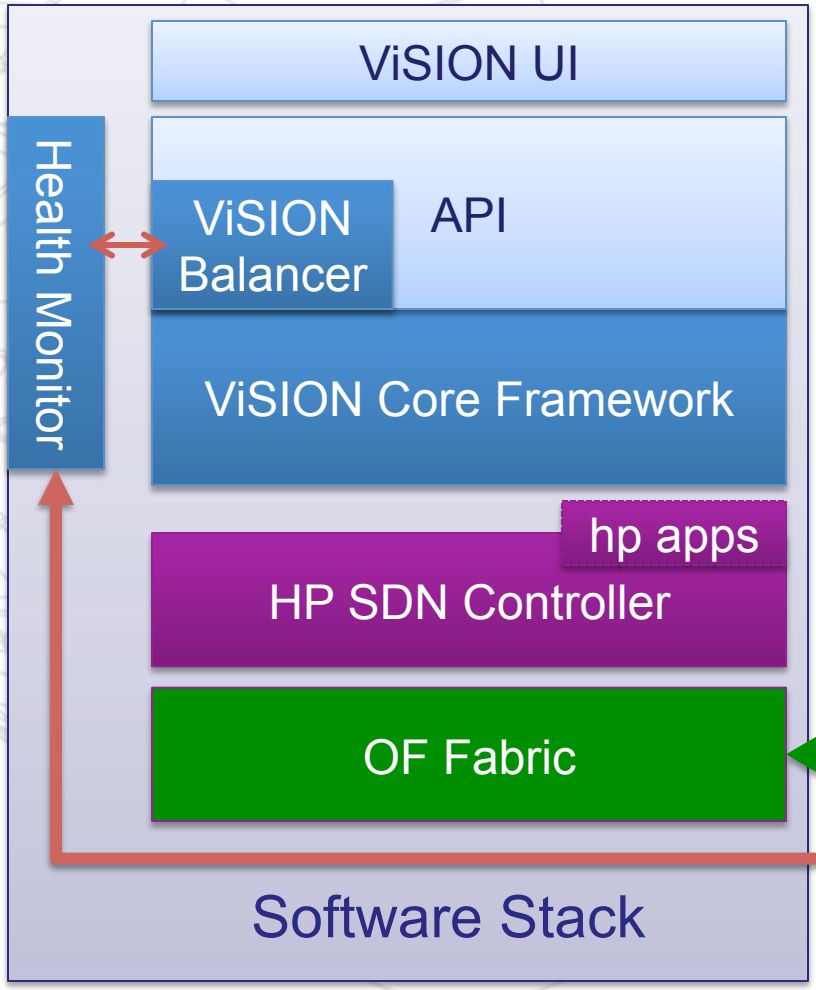
- **OpenFlow fabrics desired functionality:**
  - (1) Classification
  - (2) Load Balancing
  - (3) Mirroring
  - (4) Fault tolerance

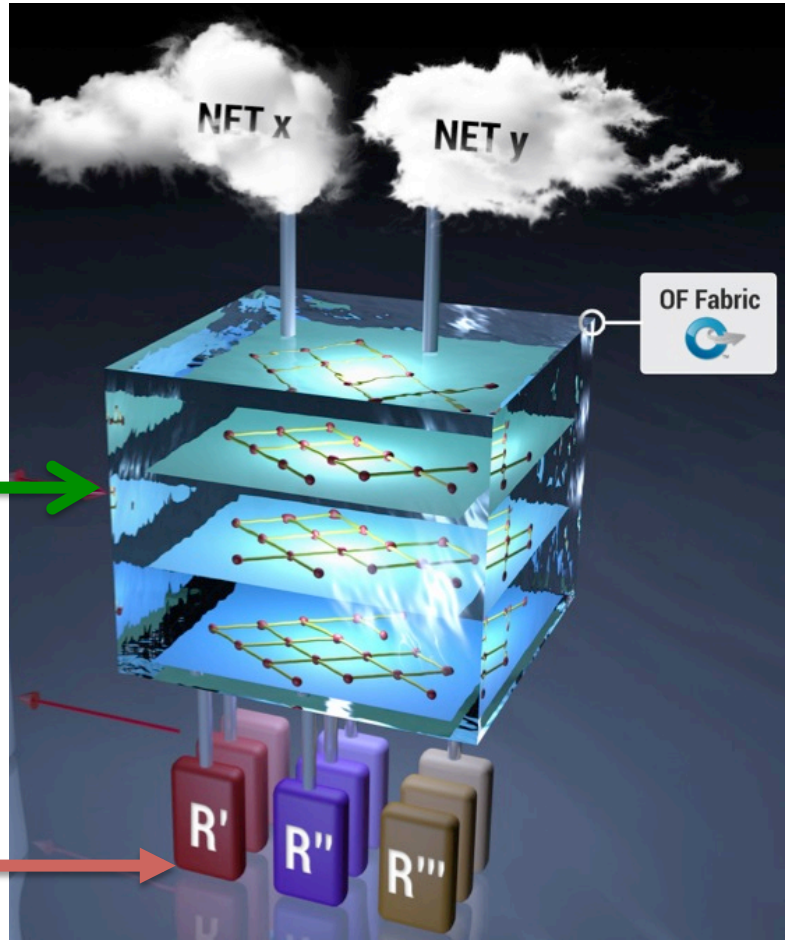
- **OpenFlow 1.0 limitations**
  - No mirroring support in early OF versions
  - Classification based on port ranges scales poorly
  - Uniform load distribution not straight forward
    - Can’t hash on high entropy bits (e.g. lower IP bits)



Background image: Shutterstock

ViSION Software Stack



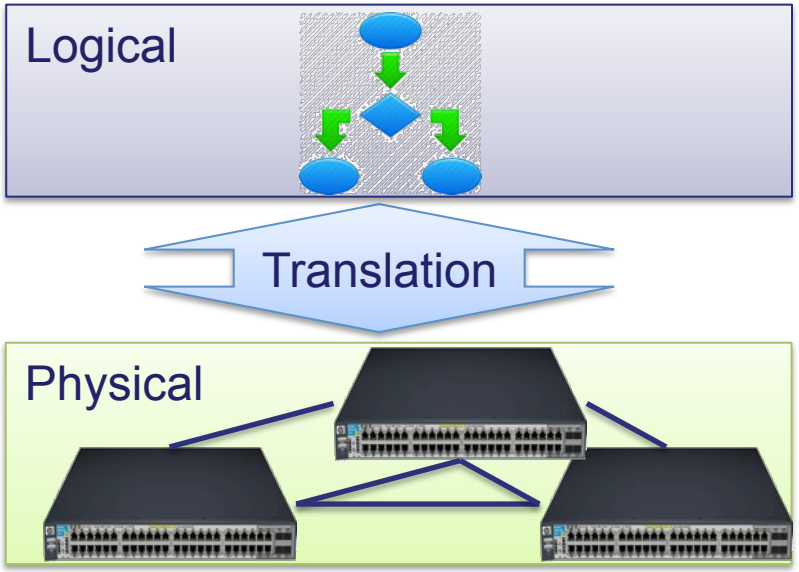


Background image: Shutterstock

ViSION/ Core Framework

- Traffic orchestration decomposition

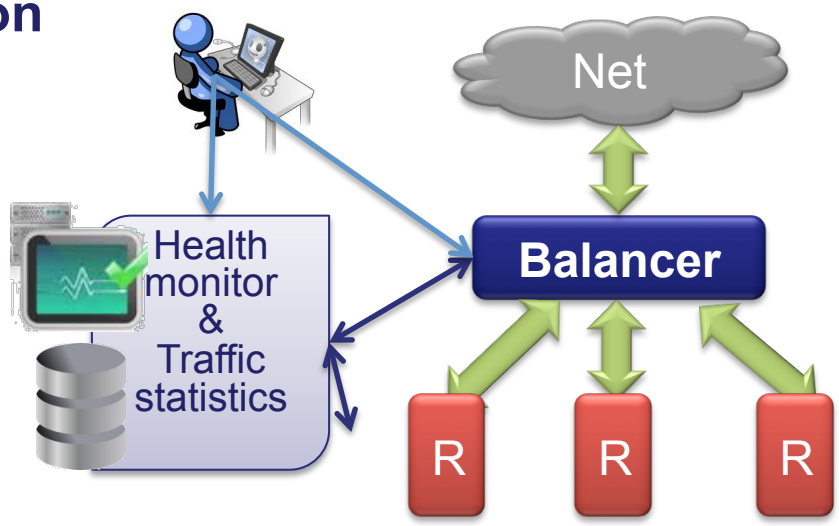
1. Logical layer: high level user goals
2. Translation layer
3. Physical layer: the OpenFlow fabric



- Core module
  - Implements the first two layers
    - Provides support for redundancy by using multiple links/paths
    - Allows the higher logical layer to focus on traffic orchestration only
  - Integrated with latest HPN Controller

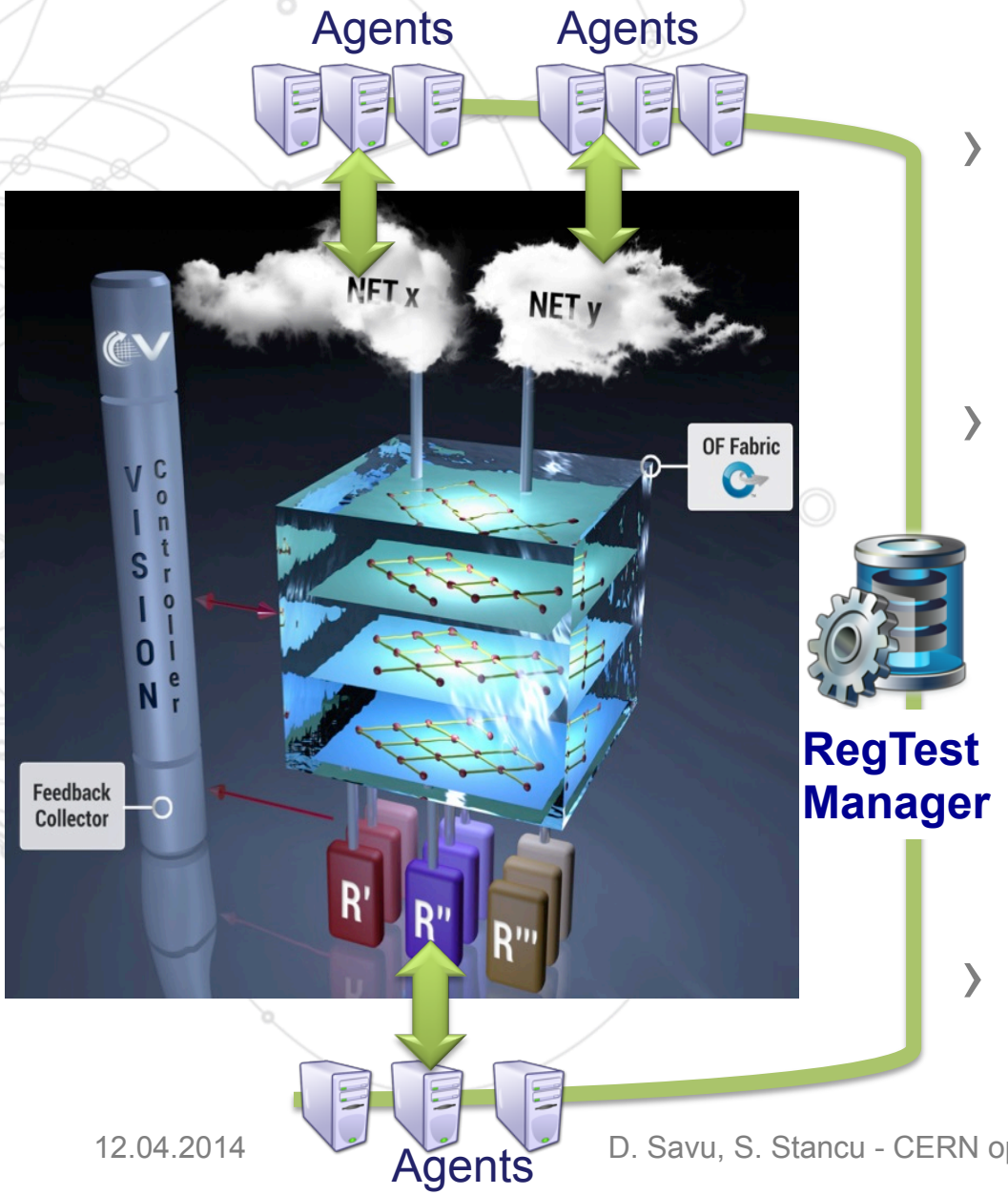
Balancer

- **Allocates flows to resources based on**
  - Resource capacity
  - Resource availability ← Health Monitor
  - Resource load ← Traffic Statistics



- **Higher level of abstraction**
  - Deals with the available resources and consumers
  - The core implements its decision into the physical OF fabric
- **Flow allocation**
  - Static → compromise for stateful resource
  - Dynamic
- **High availability**
  - Relocate flows in case a resource becomes unavailable

Regressive Testing



- **SDN applications**
  - No established validation and troubleshooting methodologies
- **RegTest application**
  - Manager:
    - Coordinates pools of agents
    - deterministic flows sequence
  - Agents
    - Coordinate and monitors flows
    - adapted MGEN to inject traffic
- **Enables regressive testing of the ViSION traffic orchestrator**

Background image: Shutterstock

References

- **Openflow;** http://archive.openflow.org/
- **ONF;** https://www.opennetworking.org/
- **SDN Central;** http://www.sdncentral.com/
- **SDN & NFV Summit;** http://sdnconference.com/

- **Open vSwitch;** http://vswitch.org/
- **Mininet;** http://mininet.org/

- **NOX, POX;** http://www.noxrepo.org/
- **Beacon, FlowVisor;** https://openflow.stanford.edu
- **Floodlight;** http://www.projectfloodlight.org/floodlight/
- **NodeFlow;** http://garyberger.net/?p=537
- **Ryu;** http://www.osrg.net/ryu/

- **RouteFlow;** https://sites.google.com/site/routeflow/

Software Defined Networking

Q & A

Dan Savu
dan.savu@cern.ch

Stefan Stancu
stefan.stancu@cern.ch