ViSION

› **13/02/2014**

Openlab Major Review
Stefan Stancu
Dan Savu

CERN openlab

Background image: Shutterstock

ViSION Project Context

- **CERN openlab - HP Networking collaboration**
  - *SDN* research and development using *OpenFlow*
  - Started in February 2012
  - 2 engineers
- **Initial project goal: Virtual services in OpenFlow networks**
  - traffic slicing and network virtualization
  - overlapping with industry (HP's VAN, NICIRA)
- **Goal reassessment: traffic orchestration**
  - Scale-out / optimize resource utilization
  - Load balancing
- **HP openlab collaboration**
  - HP's engagement in openlab phase IV ends with the ViSION project
  - In contact with HP for openlab phase V

Background image: Shutterstock

Outline

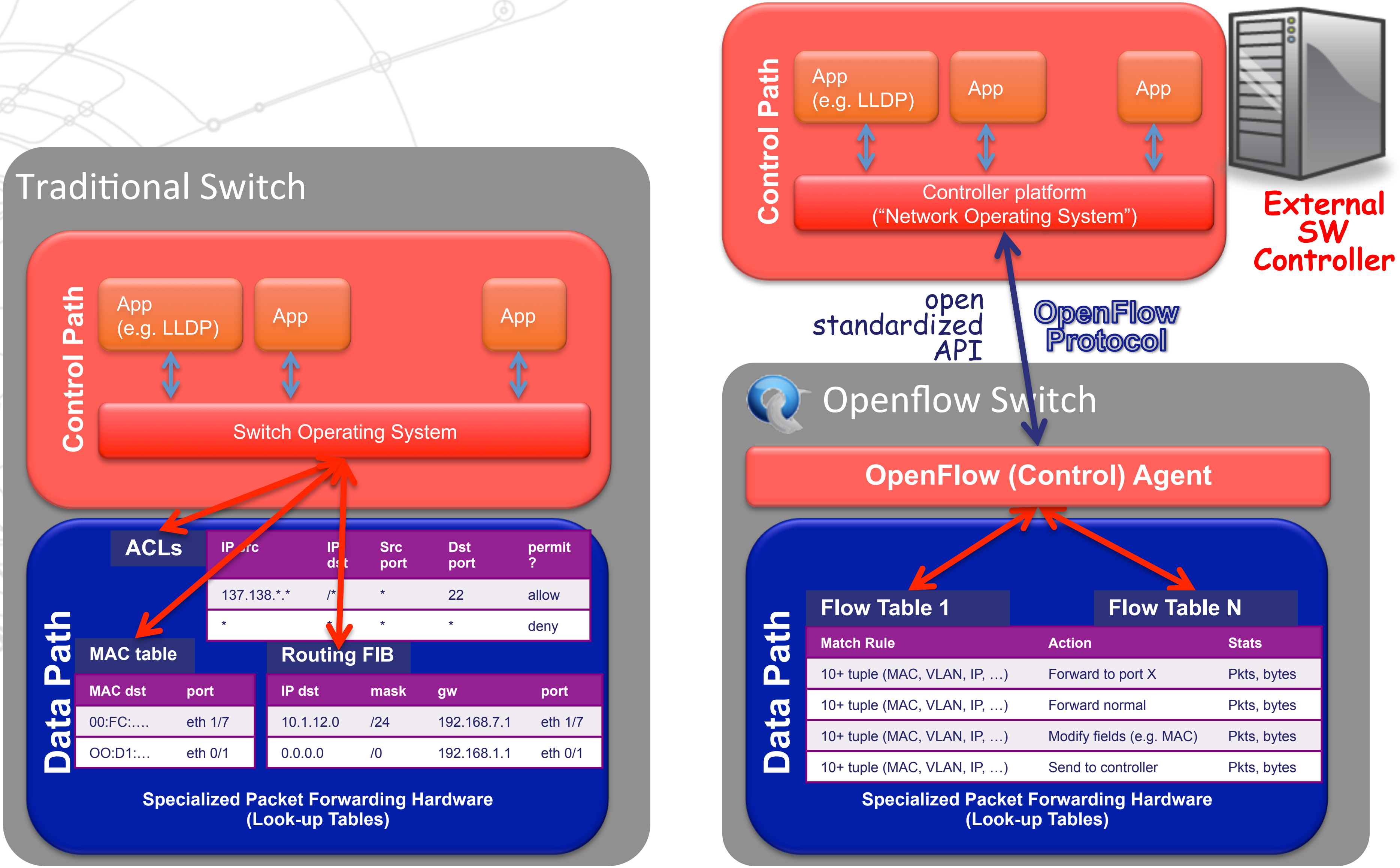
- **Software Defined Networking**
  - OpenFlow in a nutshell
  - From traditional networking to SDN
  - Protocol, Controller, Switches

- **ViSION traffic orchestrator**
  - HP SDN Controller
  - ViSION Software Stack
    - Core Framework & Balancer
    - Health Monitor
  - Regressive Testing
  - Development Environment

- **Wrap-up**
  - HPN – California
  - Project Timeline
  - Conclusion

Background image: Shutterstock

OpenFlow – Decouple Control & Data



Traditional Switch

Control Path

App (e.g. LLDP)

App

App

Switch Operating System

Data Path

ACLs

IP src	IP dst	Src port	Dst port	permit ?
137.138.*.*	/*	*	22	allow
*	*	*	*	deny

MAC table

MAC dst	port
00:FC:….	eth 1/7
OO:D1:…	eth 0/1

Routing FIB

IP dst	mask	gw	port
10.1.12.0	/24	192.168.7.1	eth 1/7
0.0.0.0	/0	192.168.1.1	eth 0/1

Specialized Packet Forwarding Hardware
(Look-up Tables)

Control Path

App (e.g. LLDP)

App

App

Controller platform
(“Network Operating System”)

External SW Controller

open standardized API

OpenFlow Protocol

Openflow Switch

OpenFlow (Control) Agent

Data Path

Flow Table 1

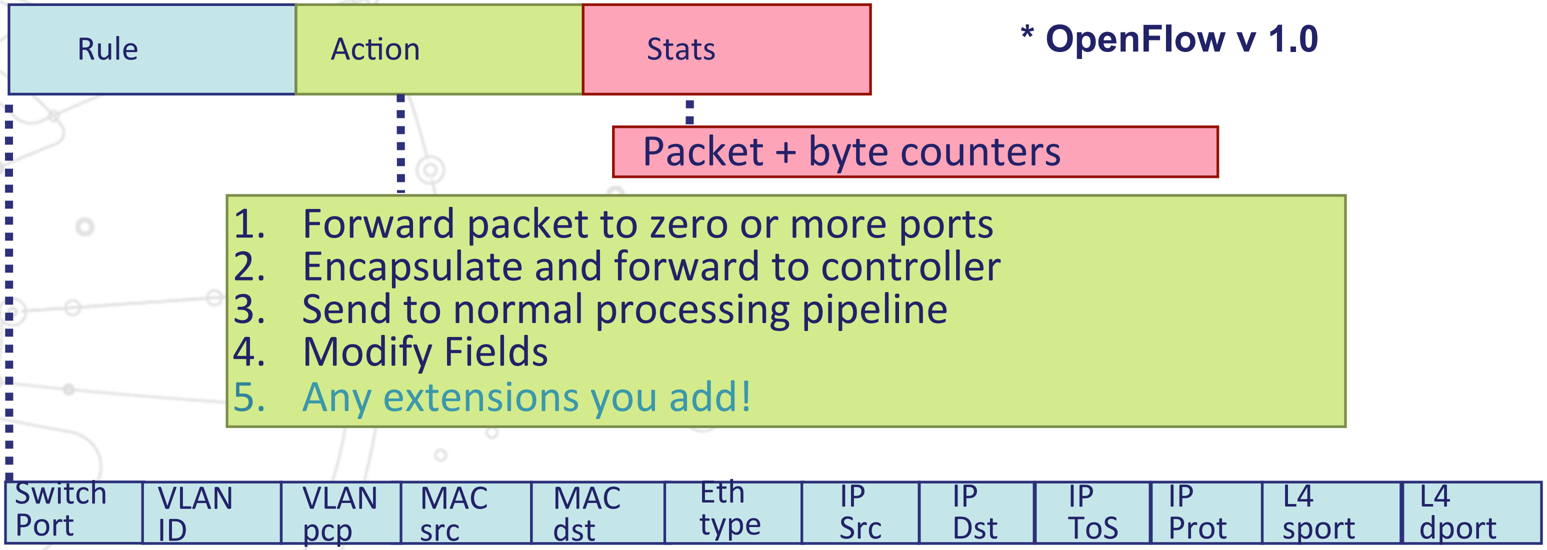
Flow Table N

Match Rule	Action	Stats
10+ tuple (MAC, VLAN, IP, …)	Forward to port X	Pkts, bytes
10+ tuple (MAC, VLAN, IP, …)	Forward normal	Pkts, bytes
10+ tuple (MAC, VLAN, IP, …)	Modify fields (e.g. MAC)	Pkts, bytes
10+ tuple (MAC, VLAN, IP, …)	Send to controller	Pkts, bytes

Specialized Packet Forwarding Hardware
(Look-up Tables)

Background image: Shutterstock

OpenFlow* – Flow Table Entries



Rule	Action	Stats

* OpenFlow v 1.0

Packet + byte counters

1. Forward packet to zero or more ports
2. Encapsulate and forward to controller
3. Send to normal processing pipeline
4. Modify Fields
5. Any extensions you add!

Switch Port	VLAN ID	VLAN pcp	MAC src	MAC dst	Eth type	IP Src	IP Dst	IP ToS	IP Prot	L4 sport	L4 dport

+ mask what fields to match
+ wildcard bits in selected fields (e.g. IP addresses)

Examples

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

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

Background image: Shutterstock

OpenFlow Evolution

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

1.0

- **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

1.1

- **02/2011**
- Multiple tables
- Multi-level VLAN and MPLS tagging
- Virtual ports

1.2

- **12/2011**
- Extensible match support (TLV)
- Basic IPv6 support
- Message bundles

1.3

- **04/2012**
  - **Next gen ASICs**
- IPv6 extension header
- Per-flow meters
- Eviction and vacancy events

1.4

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

Background image: Shutterstock

Openflow Switches *

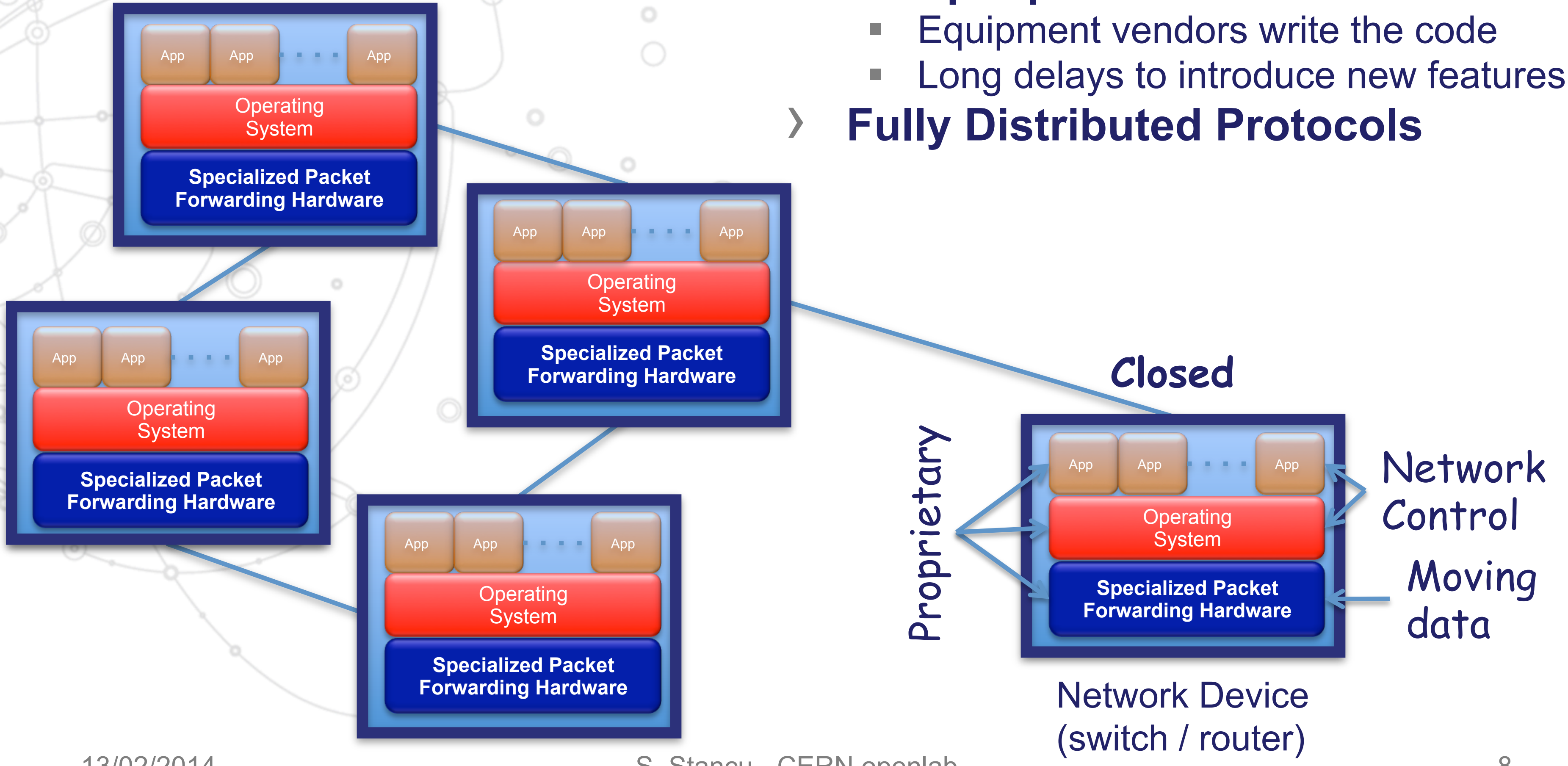
Company	Switch
HP	5400zl, 8200zl, 6200yl 3500 3500yl, 6600
Brocade	MLXe, CER, CES
Ciena	Coredirector w/ firmware 6.1.1
Cisco	Cat6k, Catlyst 3750, 6500 series
Juniper	MX, EX, T-640
Arista	EOS, 7050, 7124FX
NEC	IP8800, PF5240, PF5820
Pronto	3240, 3290, 3295, 3780
Toroki	Lightswitch 4810
Dell	Z9000, S4810
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)**

Background image: Shutterstock

Traditional Networking

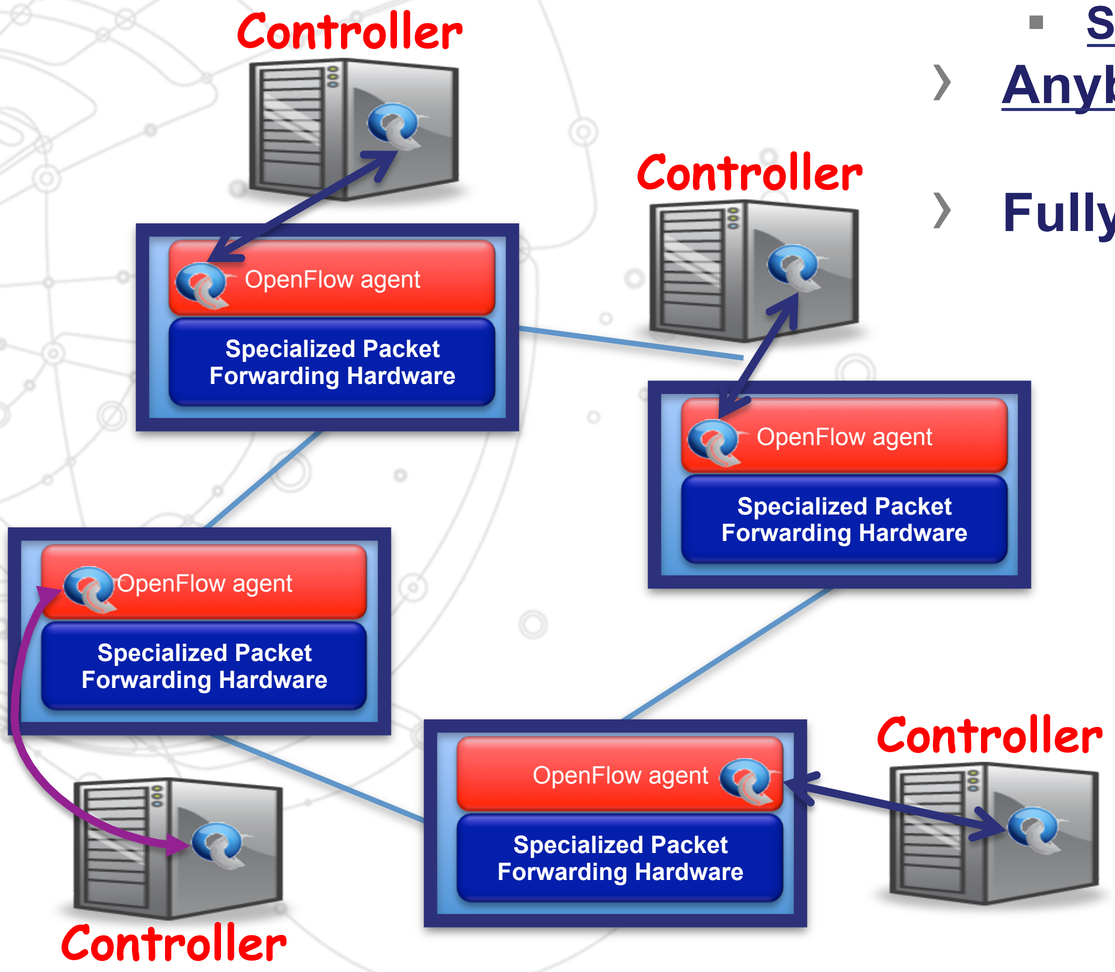
- **Closed equipment**
  - Software bundled with hardware
  - Vendor-specific interfaces
- **Few people can innovate**
  - Equipment vendors write the code
  - Long delays to introduce new features
- **Fully Distributed Protocols**


App
App
App
Operating System
Specialized Packet Forwarding Hardware
Closed
Proprietary
Network Control
Moving data
Network Device (switch / router)

Background image: Shutterstock

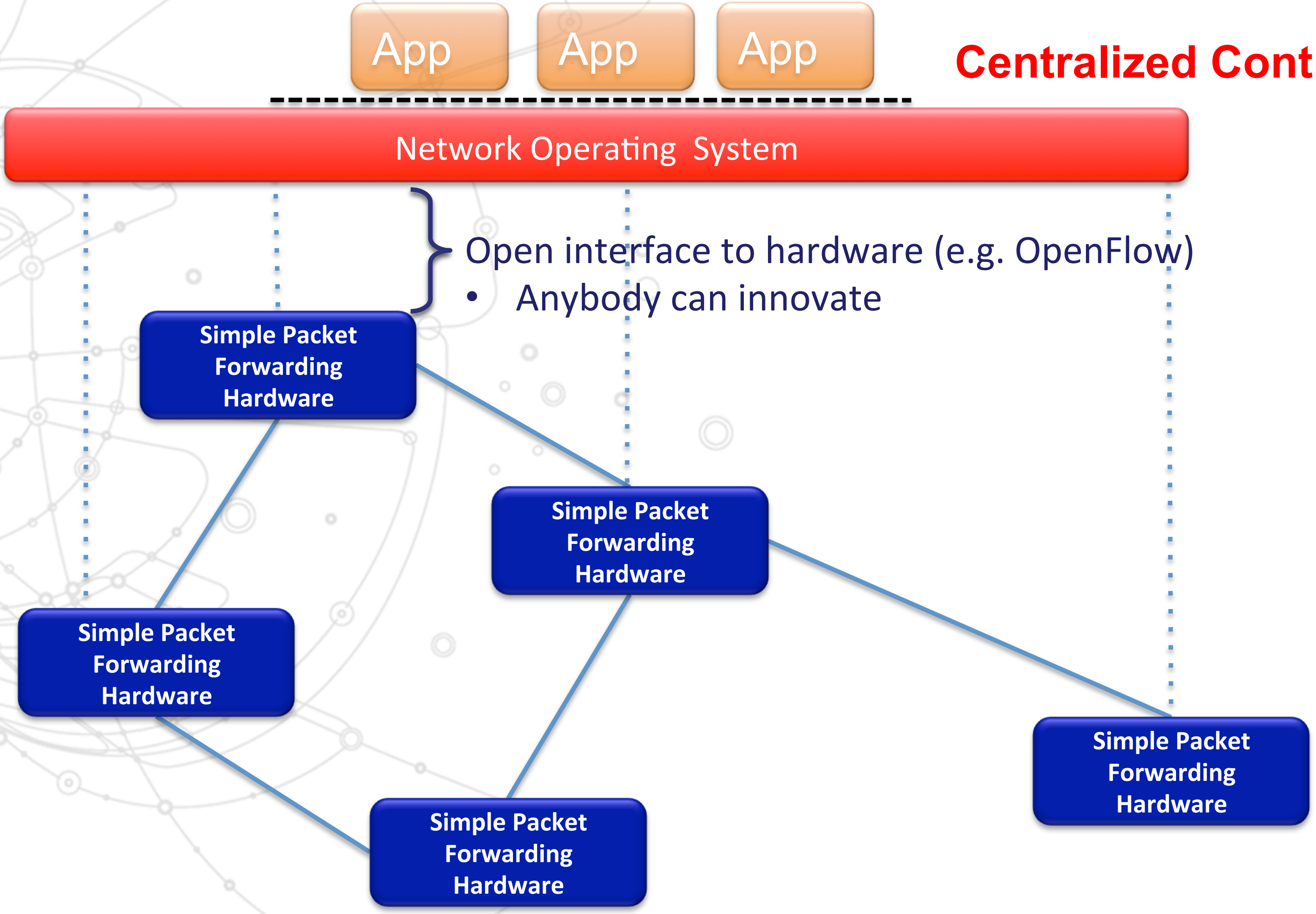
Traditional Networking ++

- <u>Open</u> equipment
  - Software **<u>decoupled</u>** from hardware
  - **<u>Standard</u>** interface (OpenFlow)
- <u>Anybody</u> can innovate
- Fully Distributed Protocols


Controller
OpenFlow agent
Specialized Packet Forwarding Hardware
Controller
OpenFlow agent
Specialized Packet Forwarding Hardware
OpenFlow agent
Specialized Packet Forwarding Hardware
Controller
OpenFlow agent
Specialized Packet Forwarding Hardware
Controller

Background image: Shutterstock

Software Defined Networking (SDN)


App
App
App
Network Operating System
Simple Packet Forwarding Hardware
Simple Packet Forwarding Hardware
Simple Packet Forwarding Hardware
Simple Packet Forwarding Hardware
Simple Packet Forwarding Hardware

Centralized Control

Open interface to hardware (e.g. OpenFlow)

- Anybody can innovate

Background image: Shutterstock

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 Release v 1.0 (02/2014
	ON.LAB ONOS (Open Networking Operating System)	Floodlight based Work in progress High availability, distributed, scale-out

Background image: Shutterstock

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

Background image: Shutterstock

Outline

- **Software Defined Networking**
  - OpenFlow in a nutshell
  - From traditional networking to SDN
  - Protocol, Controller, Switches
- **ViSION traffic orchestrator**
  - HP SDN Controller
  - ViSION Software Stack
    - Core Framework & Balancer
    - Health Monitor
  - Regressive Testing
  - Development Environment
- **Wrap-up**
  - HPN – California
  - Project Timeline
  - Conclusion

Background image: Shutterstock

HP SDN Controller Overview

Base OpenFlow Controller Appliance

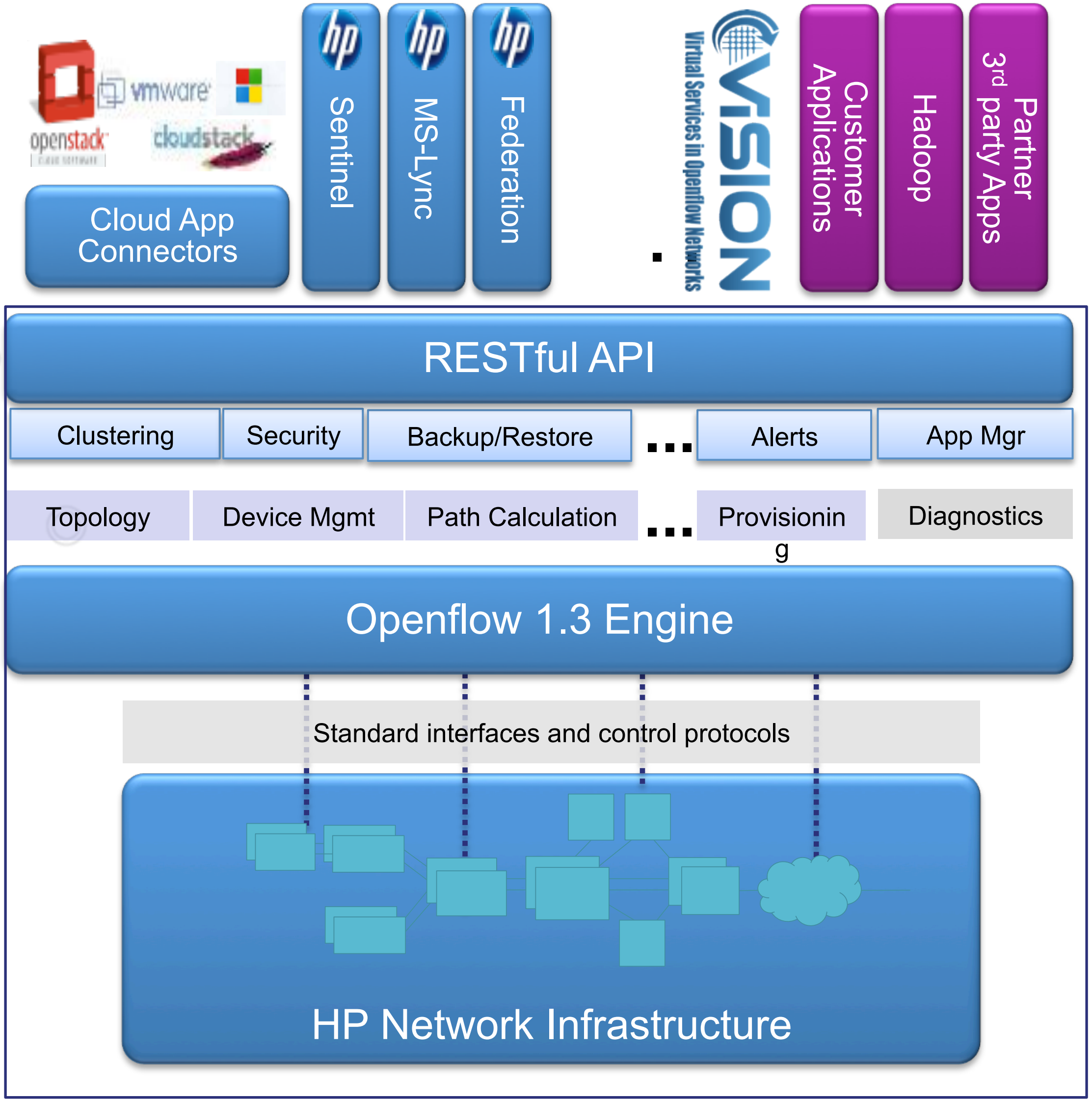
- Virtual Appliance deployed as SW on an Industry standard x86 server
- OpenFlow 1.3 Controller
- Built-in Network Services
- Appliance Administration

A Distributed Platform for High-Availability & Scalability

- Controller Clustering for Load-Balancing and Fail-Over
- Control State Mirroring across cluster for transparent failure recovery

Extensible Platform for SDN Application Developers

- Embedded Java Application Deployment
- REST APIs
- GUI



RESTful API

Clustering | Security | Backup/Restore | ... | Alerts | App Mgr

Topology | Device Mgmt | Path Calculation | ... | Provisionin g | Diagnostics

Openflow 1.3 Engine

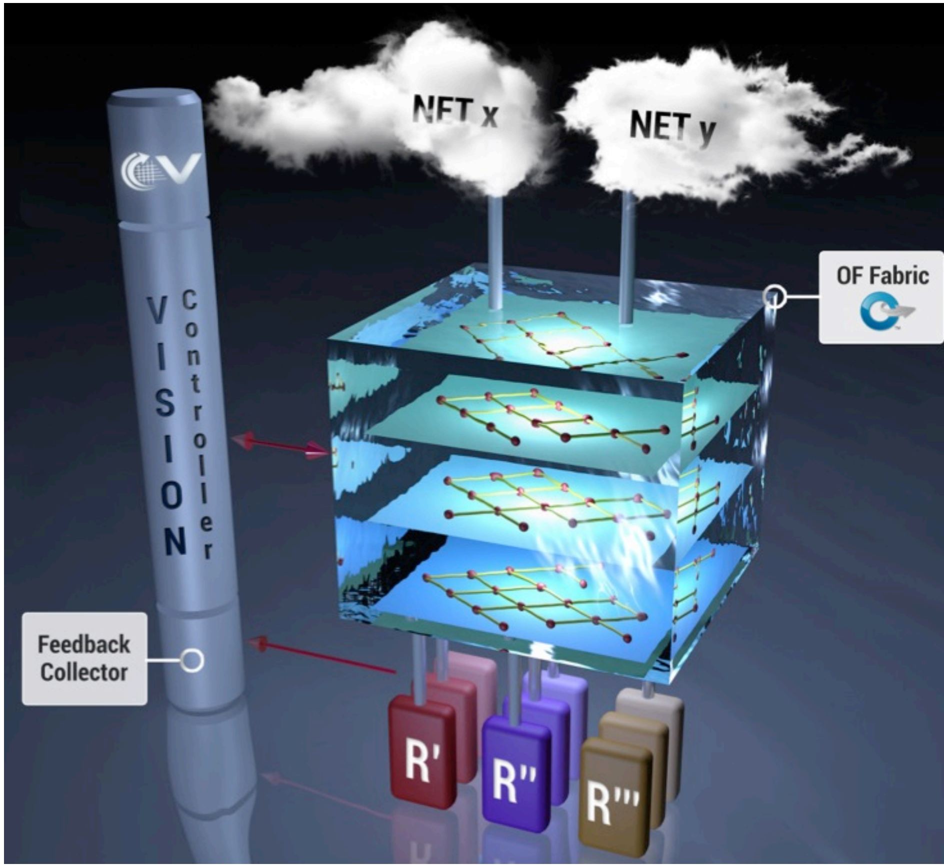
Standard interfaces and control protocols

HP Network Infrastructure

Background image: Shutterstock

ViSION Traffic Orchestrator

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

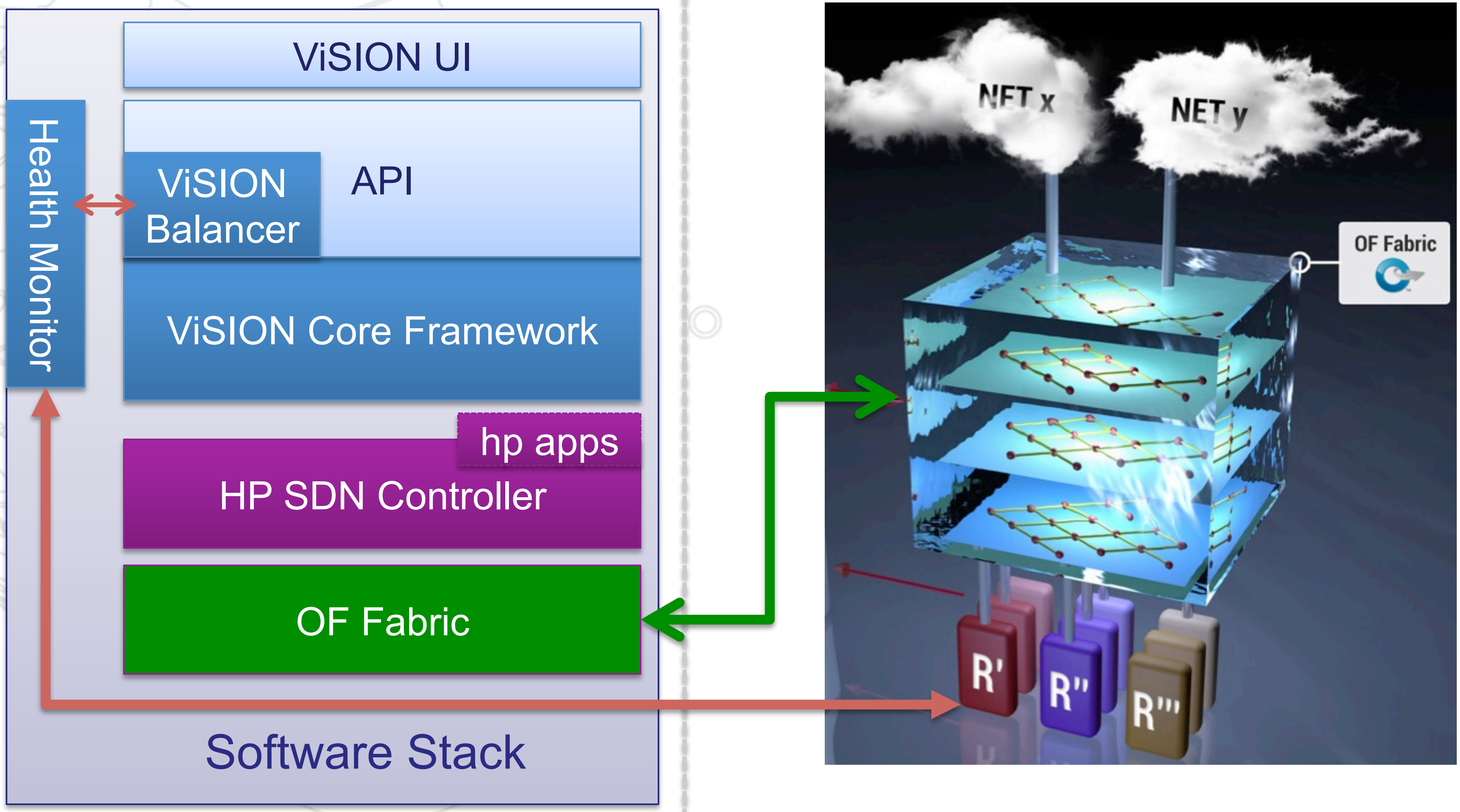


- **OpenFlow fabrics desired functionality:**
  - (1) Classification
  - (2) Load Balancing
  - (3) Mirroring
  - (4) Fault tolerance

- **OpenFlow 1.0 limitations**
  - 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

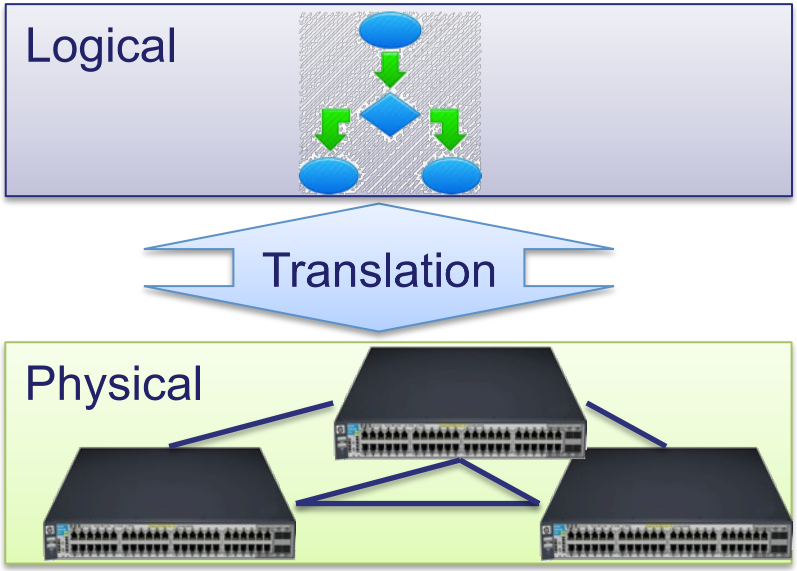


Background image: Shutterstock

ViSION/ Core Framework

- Traffic orchestration decomposition

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

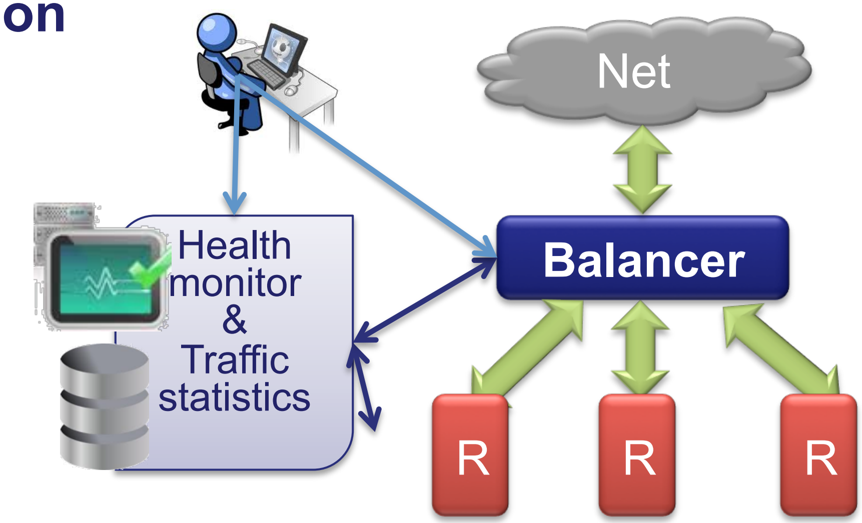


- Core module
  - Implements the first two layers
    - Provides support for redundancy by using multiple links/paths
    - Allows the higher logical layer to focus on traffic orchestration only

Background image: Shutterstock

Balancer

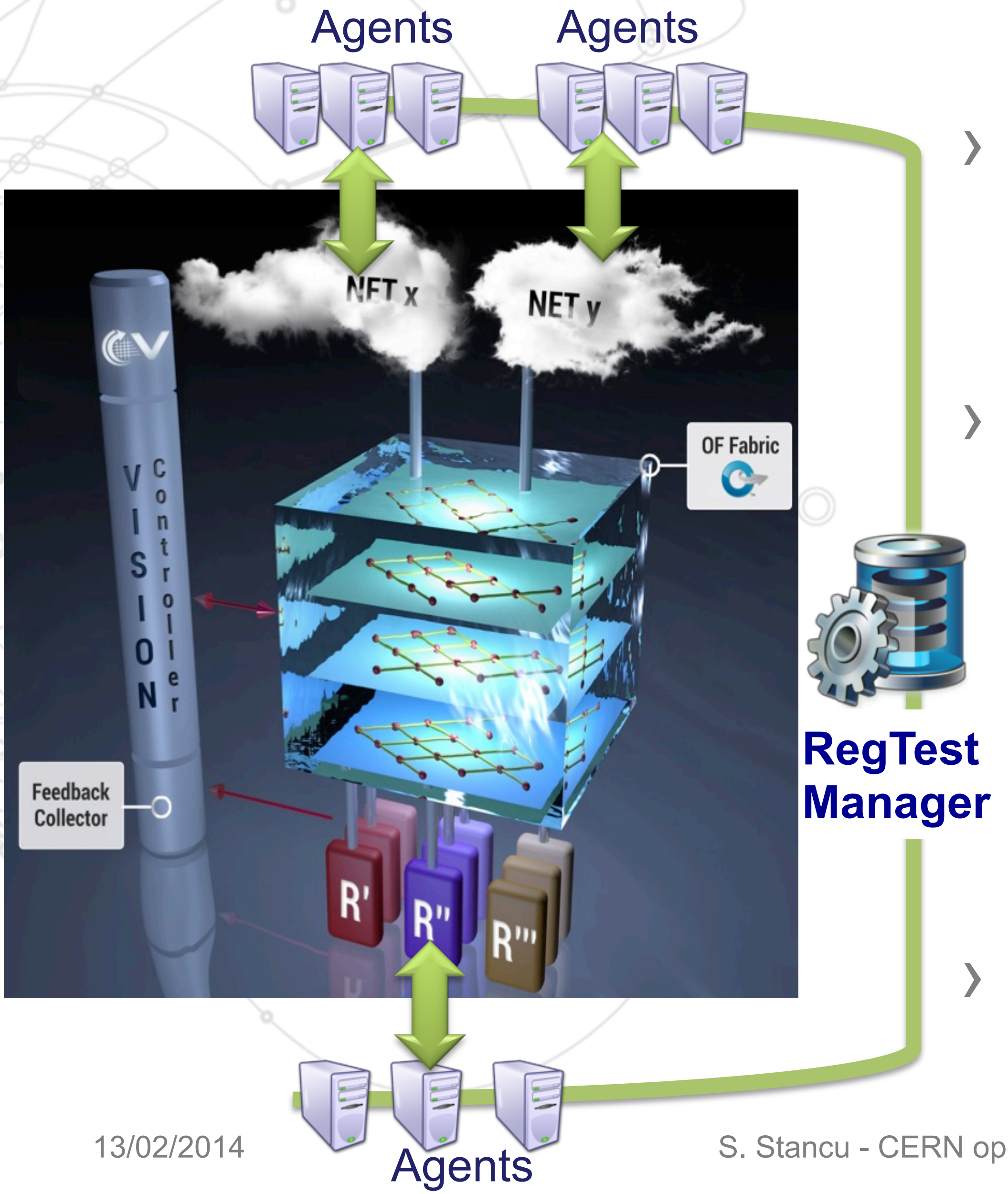
- **Allocates flows to resources based on**
  - Resource capacity
  - Resource availability ← Health Monitor
  - Resource load ← Traffic Statistics



- **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

Background image: Shutterstock

Regression 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

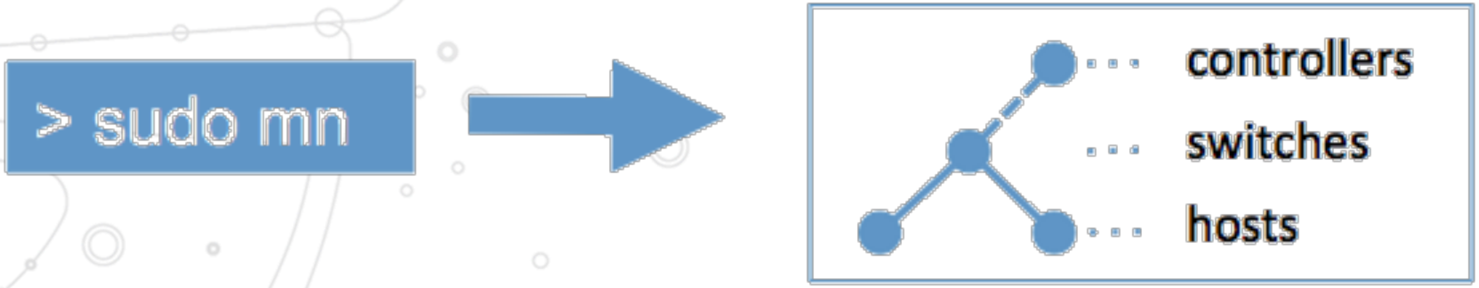
- **Enables regression testing for the ViSION traffic orchestrator**

Background image: Shutterstock

Development Environment

Mininet

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



```
> sudo mn
```

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

Background image: Shutterstock

Outline

- **Software Defined Networking**
  - OpenFlow in a nutshell
  - From traditional networking to SDN
  - Protocol, Controller, Switches
- **ViSION traffic orchestrator**
  - HP SDN Controller
  - ViSION Software Stack
    - Core Framework & Balancer
    - Health Monitor
  - Regressive Testing
  - Development Environment
- **Wrap-up**
  - HPN – California
  - Project Timeline
  - Conclusion

Background image: Shutterstock

HPN – California

HPN on-site reviews

- Architecture
  - Review and approval
  - Knowledge transfer to technology group
  - Implementation technologies assessment

- Implementation
  - Review and feedback
  - Development practices

- Brainstorming on applicability/extension to cloud environments

Background image: Shutterstock

ViSION Project Timeline

Feb 2012

- **Kick-off**
- Initial target: virtual machine mobility
- Technology review

May 2012

- **Project definition**
- Investigation
- Finalized project scope

Sep 2012

- **Design**
- OF based ACL pusher prototype
- Testbed setup

Jan 2013

- **Development**
- Core module prototype implementation
- Architecture review

May 2013

- **Development**
- **New HP controller API**
- Visit to HPN
  - In depth design review
  - Flare update

Sep 2013

- **Development**
- Service oriented core module implementation
- Health Monitor
- Regressive testing tool

Jan 2014

- **Wrap up**

Background image: Shutterstock

Conclusion

- *The ViSION core framework offers a platform for implementing traffic orchestration*

- **Outlook – CERN and HP to assess technology applicability**
  - CERN – scaling firewall system and datacenter flow optimization
  - HP – leverage the solution and know-how to expand the SDN platform.

- **Acknowledgements**
  - Thankful to HP for the excellent collaboration
    - Benefited from their pioneering experience in OpenFlow/SDN
  - Thank you to IT-CS for their support, advice and technical feedback

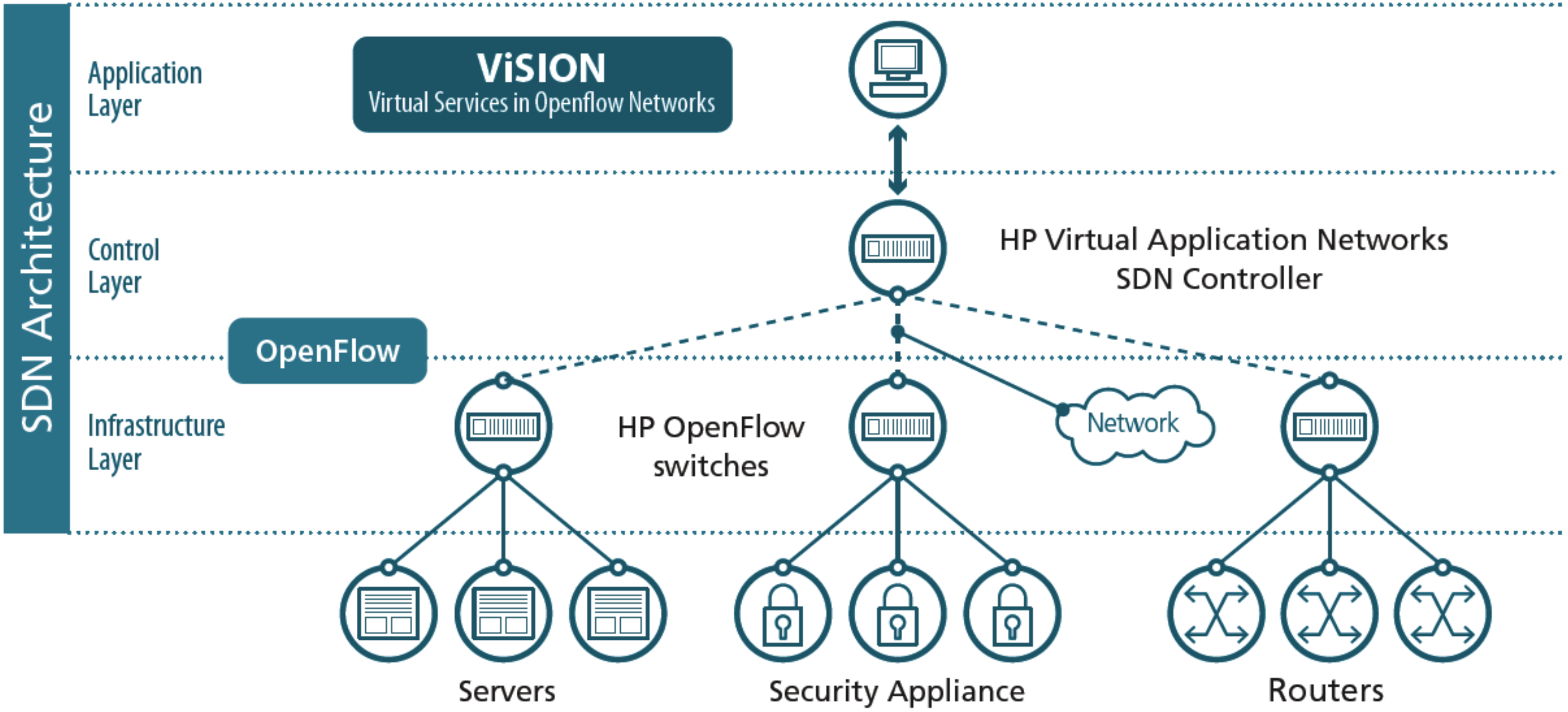
Background image: Shutterstock

Backup

Background image: Shutterstock

ViSION - HP SDN Framework

Traffic Classification | Network Resource Scaling | Load Balancing



SDN Architecture

Application Layer

ViSION
Virtual Services in Openflow Networks

Control Layer

HP Virtual Application Networks
SDN Controller

OpenFlow

Infrastructure Layer

HP OpenFlow switches

Network

Servers

Security Appliance

Routers

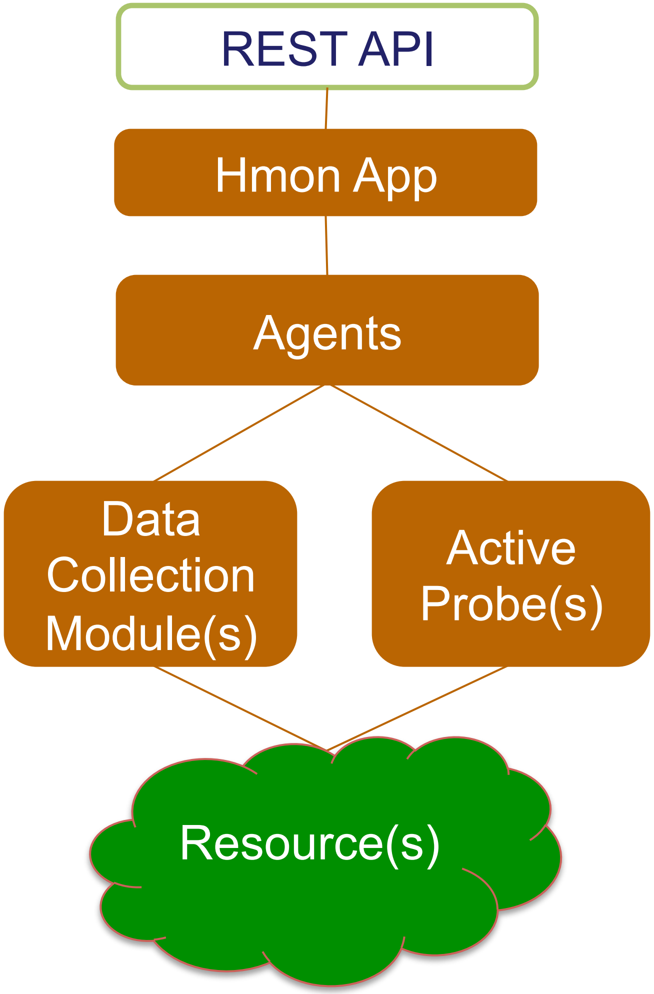
Extends scalability | Improved resource utilisation | Standards-based using OpenFlow

Background image: Shutterstock

ViSION/ Health Monitor Module

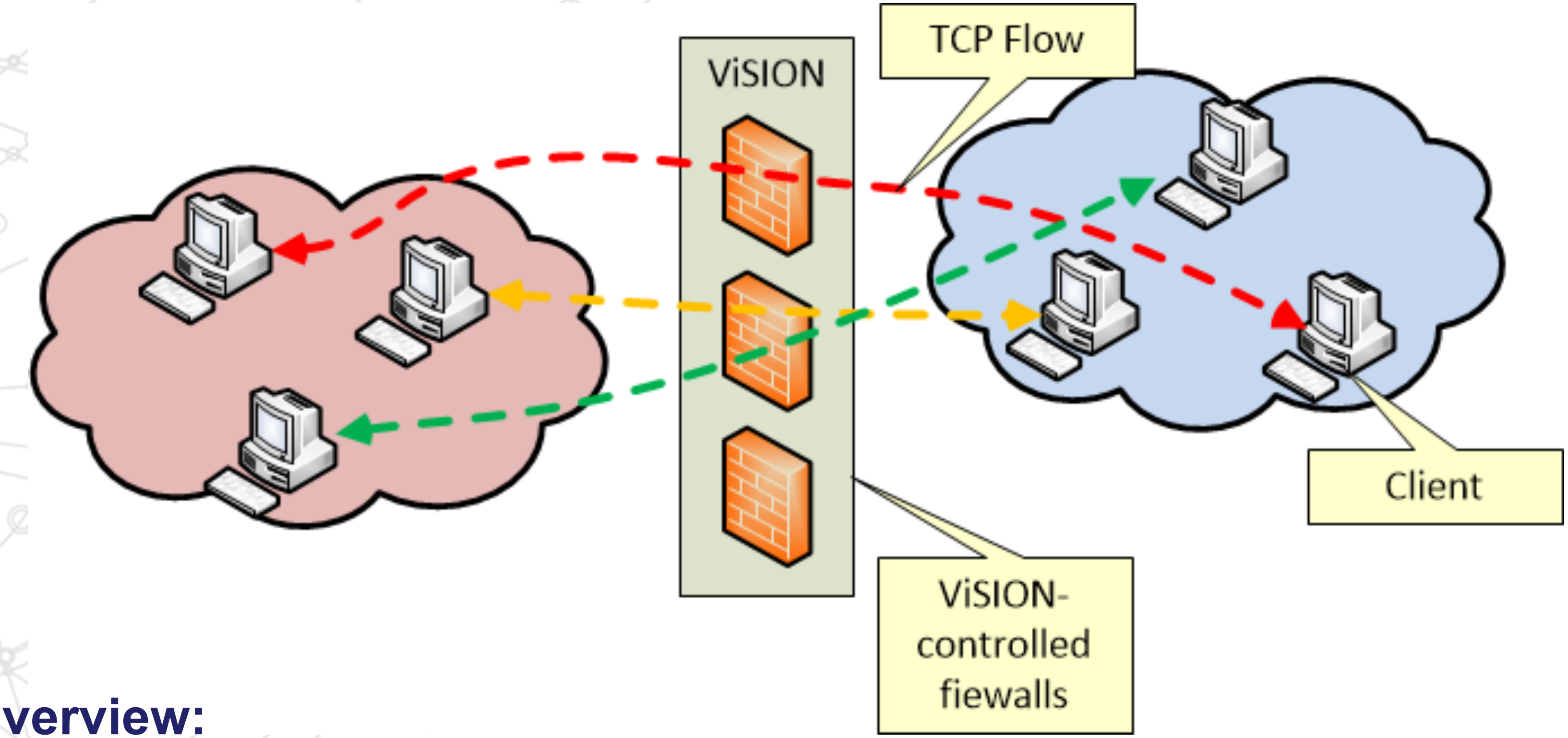
- Real-time resource status information
- Flexible module system
- Multi-module data aggregation per agent
- Custom agent configuration

- **Current modules:**
  - SNMP
  - Ping
  - HTTP Web Request
  - LinkProbe via OF packet injection



Background image: Shutterstock

ViSION Regressive Testing App



Overview:

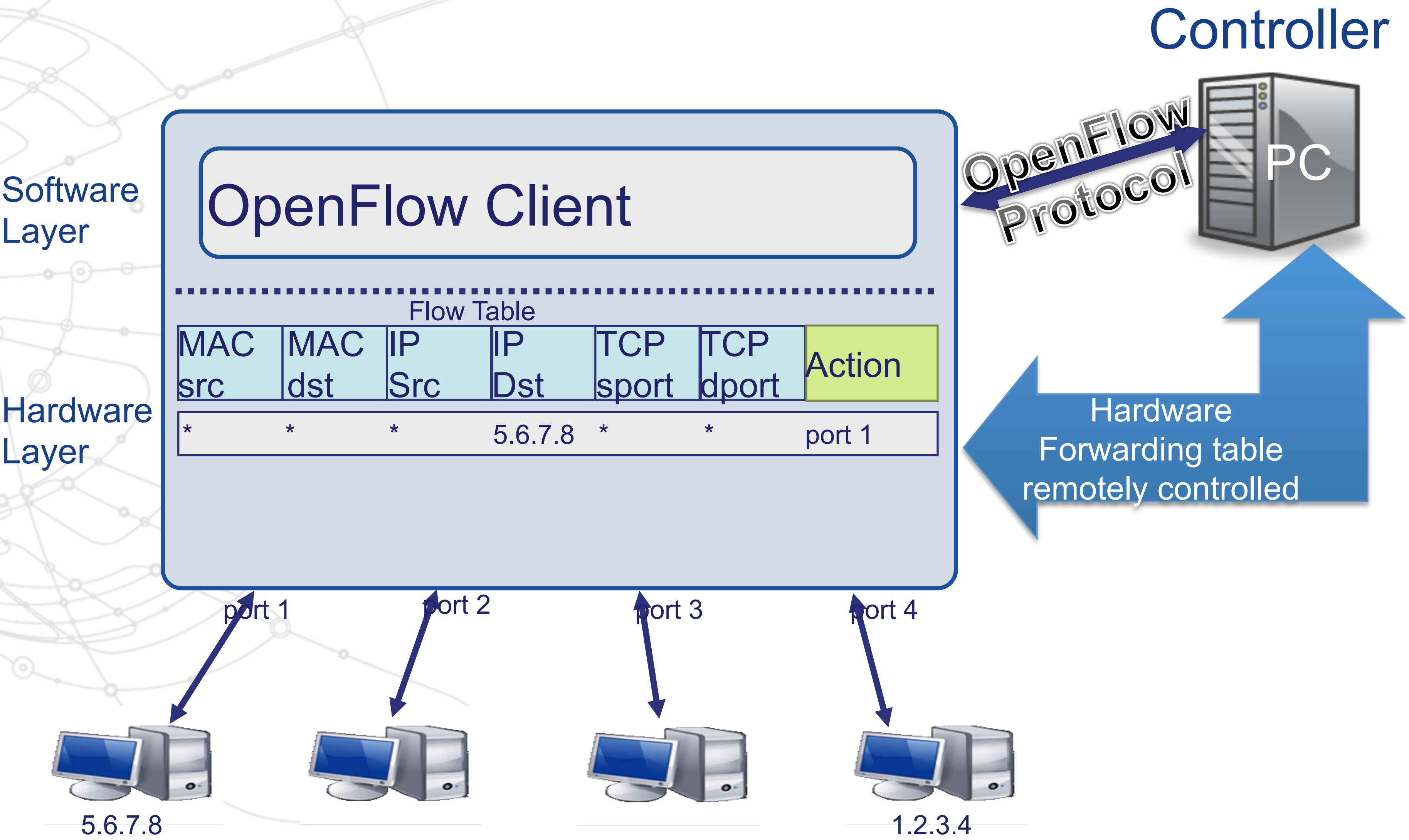
- adapted mgen used for traffic injection
- agent application to coordinate and monitor a machine’s flows
- manager application to coordinate a pool of agents

Status:

- prototype, not yet finalized
- summer student involvement

Background image: Shutterstock

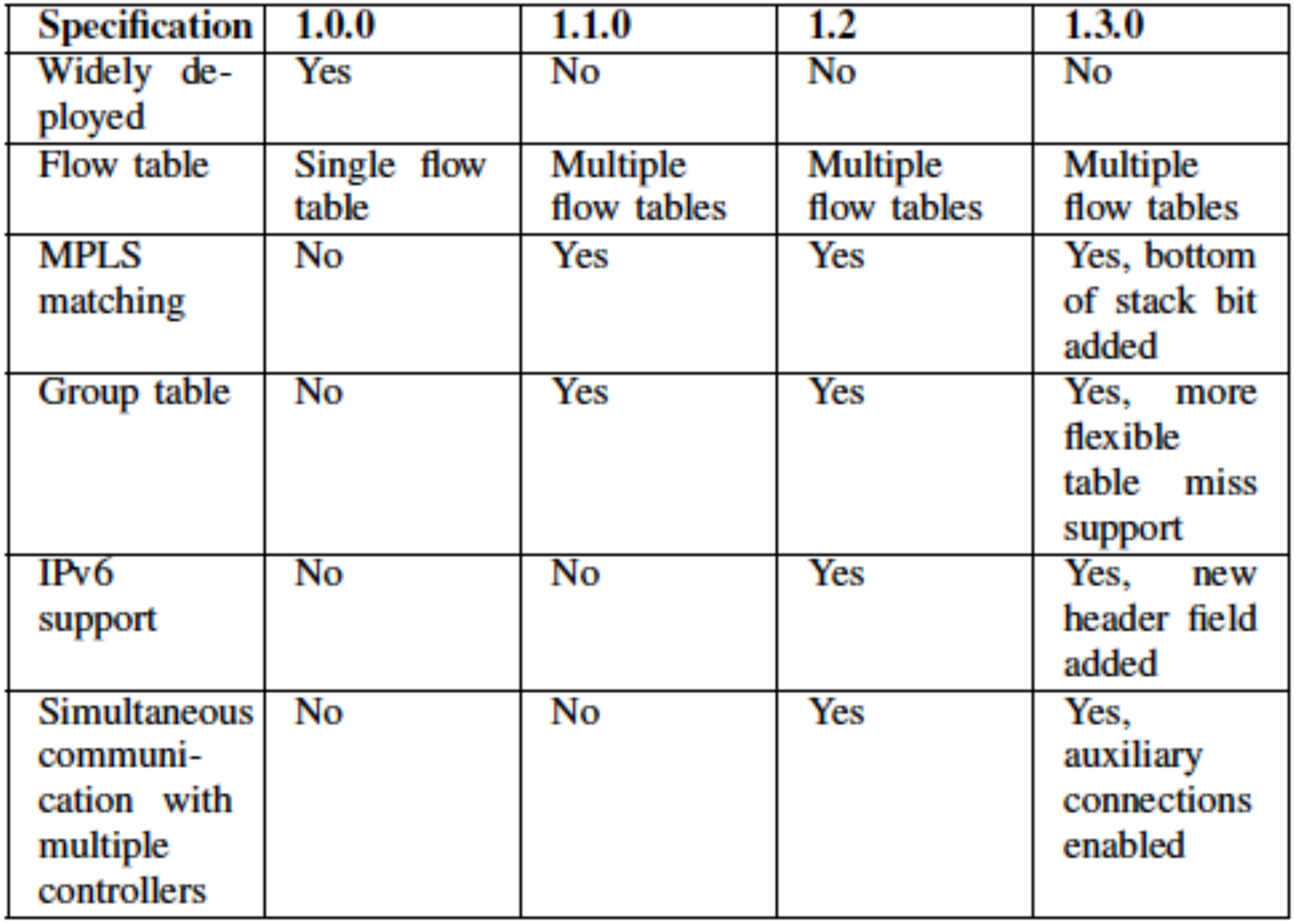
OpenFlow Example



MAC src	MAC dst	IP Src	IP Dst	TCP sport	TCP dport	Action
*	*	*	5.6.7.8	*	*	port 1

Background image: Shutterstock

OpenFlow Features Matrix



Specification	1.0.0	1.1.0	1.2	1.3.0
Widely deployed	Yes	No	No	No
Flow table	Single flow table	Multiple flow tables	Multiple flow tables	Multiple flow tables
MPLS matching	No	Yes	Yes	Yes, bottom of stack bit added
Group table	No	Yes	Yes	Yes, more flexible table miss support
IPv6 support	No	No	Yes	Yes, new header field added
Simultaneous communication with multiple controllers	No	No	Yes	Yes, auxiliary connections enabled

Background image: Shutterstock

Openflow Evolution

- **0.2 (Mar 2008)**
- **1.0 (Dec 2009):**
  - **match: input port, vlan, mac, ip, transport ports; actions: port ouput, set/strip vlan, set mac, ip, transport ports**; slicing (multiple queues per output port); flow cookie; datapath description; match on ip in arp packets; port stats; TLS controller communication; explicit IP fragmentation; **subnet masks**; IN_PORT; port and link status events; **vendor extensions**; spanning tree access; modify actions in existing flows; ICMP match; **controller failover; barrier command;** emergency flows for controller lost handling; VLAN priority match.
- **1.1 (Feb 2011):**
  - **Multiple tables;** port groups; **multi-level VLAN and MPLS tagging;** virtual ports (lag, tunnel); controller connection failure.
- **1.2 (Dec 2011):**
  - **extensible match support (TLV); basic ipv6 support;** controller role change mechanism, message bundles.
- **1.3 (Apr 2012):**
  - Flexible capabilities negotiations and table miss support; **ipv6 extension header; per-flow meters;** per connection event filtering; auxiliary connections; MPLS BoS match; tunnel-id metadata (e.g. GRE); packet-in cookies; **eviction and vacancy events.**
- **1.4 (Aug 2013):**
  - **Optical port properties;** controller flow monitoring and status events; synchronized tables; **IANA TCP port 6653 for Openflow.**

Background image: Shutterstock

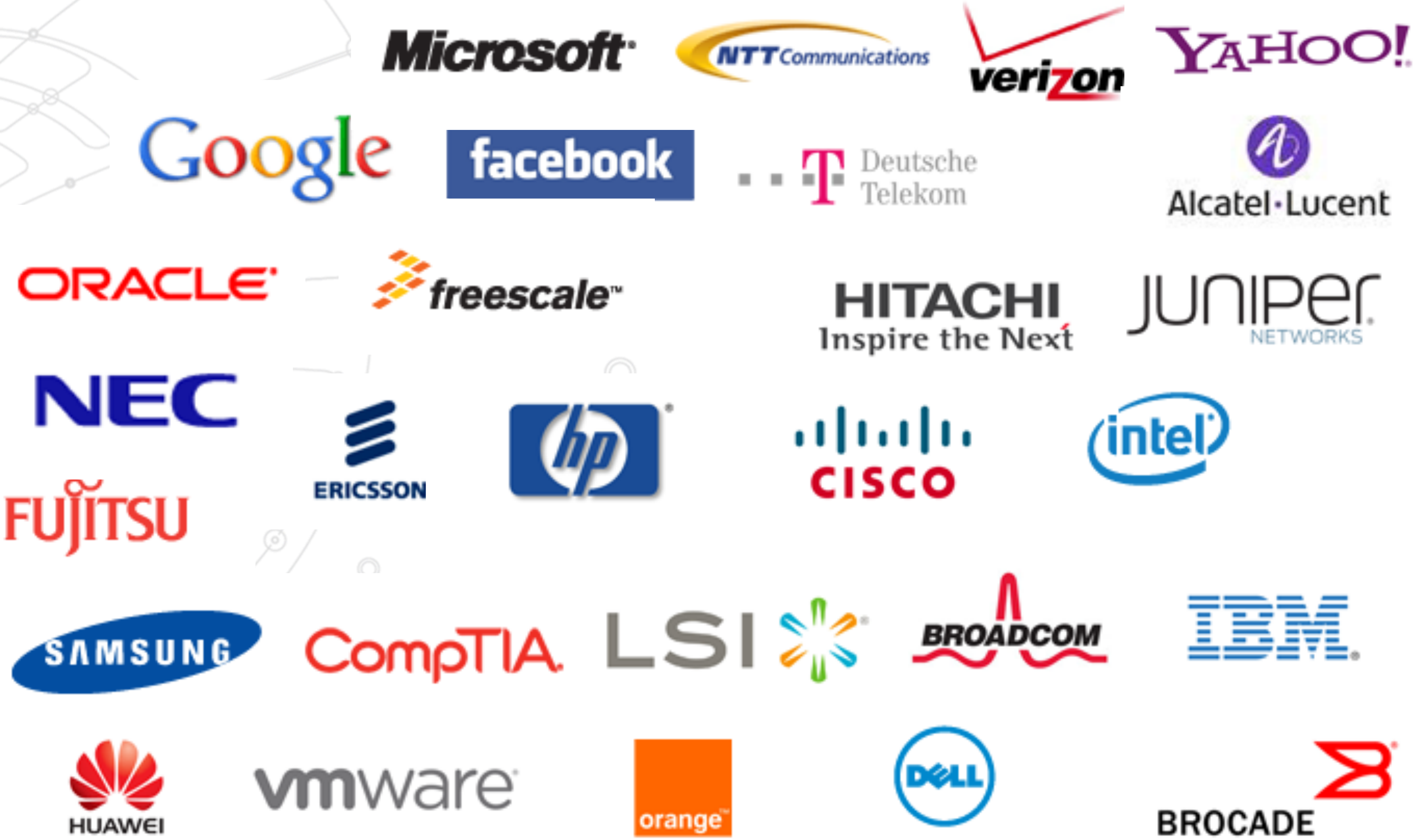
Technologies

- **Open vSwitch**
  - http://vswitch.org/
  - Production quality virtual switch, **openflow**, vlan isolation, qos, monitoring, automated control (e.g. multi-server virtualized environment, development and testing etc)
  - Part of linux kernel as of 3.3, the default switch in Xen Cloud Platform, integrated in Openstack etc
- **Openflow reference**
  - http://archive.openflow.org/wp/downloads/
- **NOX -> POX**
  - http://www.noxrepo.org/pox/about-pox/
  - C/ Phython openflow controller
- **Beacon -> Floodlight**
  - https://openflow.stanford.edu/display/Beacon/Home
- **Nodeflow (js, node.js)**
  - http://garyberger.net/?p=537
- **Routeflow**
  - https://sites.google.com/site/routeflow/
- **Oflops**
  - http://archive.openflow.org/wk/index.php/Oflops
  - Controller to benchmark openflow switches
- **RYU SDN framework, python**
  - http://www.osrg.net/ryu/
- **Flowvisor , network slicing**
  - https://github.com/OPENNETWORKINGLAB/flowvisor/wiki
- **STS, SDN Troubleshooting Simulator**
  - http://ucb-sts.github.io/sts/

http://yuba.stanford.edu/~casado/of-sw.html

Background image: Shutterstock

ONF Members



Background image: Shutterstock