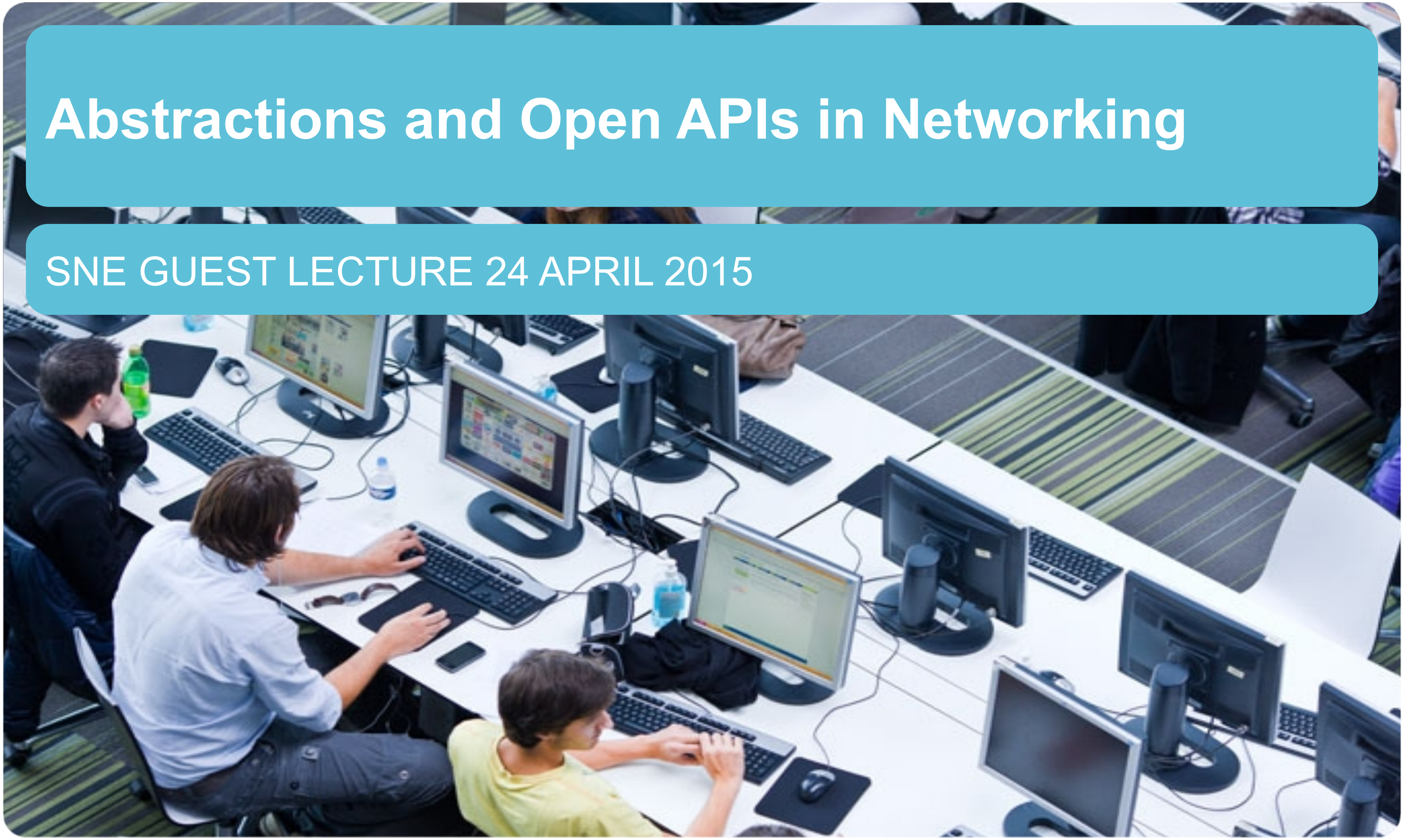
Abstractions and Open APIs in Networking

SNE GUEST LECTURE 24 APRIL 2015



Ronald van der Pol <Ronald.vanderPol@SURFnet.nl>

SURF NET

Outline

Abstractions

Open Hardware & Open Compute Project

Software Defined Networking & OpenFlow

ONOS & OpenDaylight SDN Controller Frameworks

Google & OpenFlow

SURFnet OpenFlow project

Open vSwitch & Network Functions Virtualisation (NFV)

Summary

Current Networking Practice

35 year years of thinking in layers (e.g. OSI model)

Layer N+1 encapsulates packets/frames from layer N

Each layer is just adding a header, hardly any abstractions

Application programmers still need to deal with DNS and IP addresses

Abstractions in Computing

CPUs → Assembly → higher languages → modules and classes
very few are coding in assembly

Mechanical disk drive → Device driver → Filesystem
no user is sending commands to directly move the disk head

LCD → video driver → graphical library → browser
no user is directly writing pixels to an LCD screen

How do we manage networks?

Login on the CLI of a router/switch

Type low level commands to router/switch

Configure in detail what protocols must do (set parameters)

This needs to be done for every OSI layer
- set MTU, flow control, etc
- set IP addresses
- configure routing/switching protocol

And on each and every router/switch

“Protocol Soup”

Current way to handle new functionality in networking is to define a new protocol.

A protocol is just adding a header, it is NOT hiding complexity

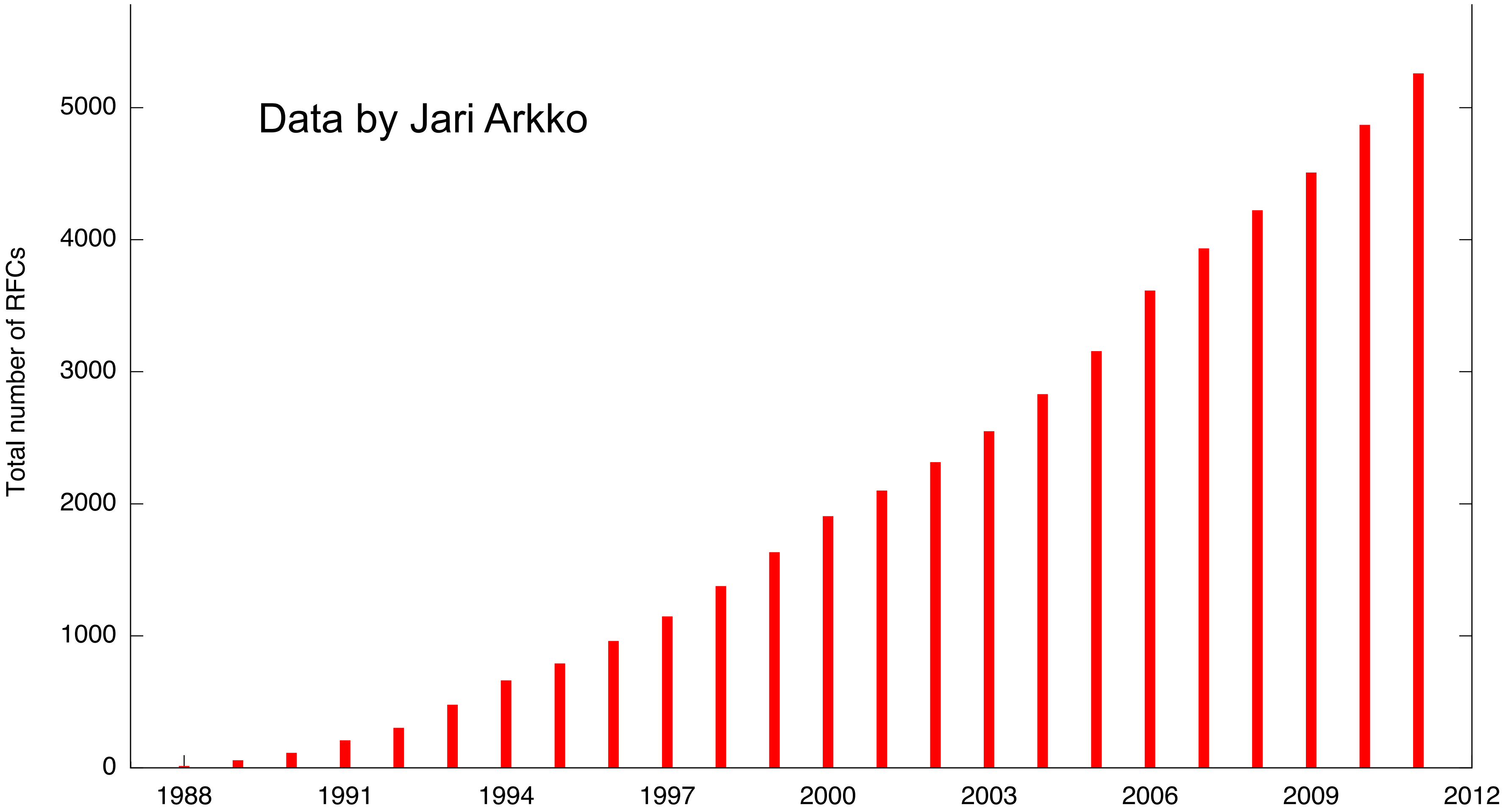
Exponential growth in network protocol standards.

Standards seem to become larger and more complex.

Vendors implement all standards, which increases costs and decreases stability.

Do you need all those standards?

Total Number of RFCs Published



IEEE 802.1Q

Simple VLAN standard?

Not really, original version amended by at least 14 additional standards.

802.1Q-1998 had 211 pages.

802.1Q-2011 has 1365 pages, and includes:
802.1u, 802.1v, 802.1s (multiple spanning trees), 802.1ad (provider bridging), 802.1ak (MRP, MVRP, MMRP), 802.1ag (CFM), 802.1ah (PBB), 802.1ap (VLAN bridges MIB), 802.1Qaw, 802.1Qay (PBB-TE), 802.1aj, 802.1Qav, 802.1Qau (congestion management), 802.1Qat (SRP)

Specs of a Modern Ethernet Switch (random example, but they are all the same)

Area Networks
- IEEE 802.3ad Static load sharing configuration and LACP based dynamic configuration
- Software Redundant Ports
- IEEE 802.1AB – LLDP Link Layer Discovery Protocol
- LLDP Media Endpoint Discovery (LLDP-MED), ANSI/TIA-1057, draft 08
- Extreme Discovery Protocol (EDP)
- Extreme Loop Recovery Protocol (ELRP)
- Extreme Link State Monitoring (ELSM)
- IEEE 802.1ag L2 Ping and traceroute, Connectivity Fault Management
- ITU-T Y.1731 Frame delay measurements

Management and Traffic Analysis

- RFC 2030 SNTP, Simple Network Time Protocol v4
- RFC 854 Telnet client and server
- RFC 783 TFTP Protocol (revision 2)
- RFC 951, 1542 BootP
- RFC 2131 BOOTP/DHCP relay agent and DHCP server
- RFC 1591 DNS (client operation)
- RFC 1155 Structure of Management Information (SMIv1)
- RFC 1157 SNMPv1
- RFC 1212, RFC 1213, RFC 1215 MIB-II,

- XML APIs over Telnet/SSH and HTTP/HTTPS
- Web-based device management interface – ExtremeXOS ScreenPlay™
- IP Route Compression

Security, Switch and Network Protection

- Secure Shell (SSH-2), Secure C… and SFTP client/server with enc… authentication (requires export … encryption module)
- SNMPv3 user based security, w… encryption/authentication (see …
- RFC 1492 TACACS+
- RFC 2138 RADIUS Authenticati…
- RFC 2139 RADIUS Accounting
- RFC 3579 RADIUS EAP support
- RADIUS Per-command Authentic…
- Access Profiles on All Routing P…
- Access Policies for Telnet/SSH-…
- Network Login – 802.1x, Web a… MAC-based mechanisms
- IEEE 802.1x – 2001 Port-Based Access Control for Network Logi…
- Multiple supplicants with multip… Network Login (all modes)
- Fallback to local authentication (MAC and Web-based methods)

Sping, Ascend, Stream, Land, Octopus

Security, Router Protection

- RFC 1587 OSPF NSSA Option
- RFC 1765 OSPF Database Overflow
- RFC 2370 OSPF Opaque LSA Option
- RFC 3623 OSPF Graceful Restart
- RFC 1850 OSPFv2 MIB
- RFC 2362 PIM-SM (Edge-mode)
- RFC 2934 PIM MIB
- RFC 3569, draft-ietf-ssm-arch-06.txt PIM-SSM PIM Source Specific Multicast
- draft-ietf-pim-mib-v2-o1.txt
- Mtrace, a "traceroute" facility for IP Multicast: draft-ietf-idmr-traceroute-ipm-07
- Mrinfo, the multicast router information tool based on Appendix-B of draft-ietf-idmr-dvmrp v3-11

IPv6 Host Services

- RFC 3587, Global Unicast Address Format
- Ping over IPv6 transport
- Traceroute over IPv6 transport
- RFC 5095, Internet Protocol, Version 6 (IPv6) Specification
- RFC 4861, Neighbor Discovery for IP Version 6, (IPv6)
- RFC 2463, Internet Control Message Protocol (ICMPv6) for the IPv6 Specification
- RFC 2464, Transmission of IPv6 Packets over Ethernet Networks
- RFC 2465, IPv6 MIB, General Group and Textual Conventions
- RFC 2466, MIB for ICMPv6
- RFC 2462, IPv6 Stateless Address Auto Configuration – Host Requirements
- RFC 1981, Path MTU Discovery for IPv6, August 1996 – Host Requirements
- RFC 3513, Internet Protocol Version 6 (IPv6) Addressing Architecture
- Telnet server over IPv6 transport
- SSH-2 server over IPv6 transport

IPv6 Interworking and Migration

- RFC 2893, Configured Tunnels
- RFC 3056, 6to4

IPv6 Router Services

- RFC 2462, IPv6 Stateless Address Auto Configuration – Router Requirements
- RFC 1981, Path MTU Discovery for IPv6, August 1996 – Router Requirements
- RFC 2710, IPv6 Multicast Listener Discovery v1 (MLDv1) Protocol
- Static Unicast routes for IPv6
- RFC 2080, RIPng

- RFC 1771 Border Gateway Protocol 4
- RFC 1965 Autonomous System Confederations for BGP
- RFC 2796 BGP Route Reflection (supersedes RFC 1966)
- RFC 1997 BGP Communities Attribute
- RFC 1745 BGP4/IDRP for IP-OSPF Interaction
- RFC 2385 TCP MD5 Authentication for BGPv4
- RFC 2439 BGP Route Flap Damping
- RFC 2918 Route Refresh Capability for BGP-4
- RFC 3392 Capabilities Advertisement with BGP-4
- RFC 4360 BGP Extended Communities Attribute
- RFC 4486 Subcodes for BGP Cease Notification message
- draft-ietf-idr-restart-10.txt Graceful Restart Mechanism for BGP
- RFC 4760 Multiprotocol extensions for BGP-4
- RFC 1657 BGP-4 MIB
- RFC 4893 BGP Support for Four-Octet AS Number Space
- Draft-ietf-idr-bgp4-mibv2-02.txt – Enhanced BGP-4 MIB
- RFC 1195 Use of OSI IS-IS for Routing in TCP/IP and Dual Environments (TCP/IP transport only)
- RFC 2763 Dynamic Hostname Exchange Mechanism for IS-IS
- RFC 2966 Domain-wide Prefix Distribution with Two-Level IS-IS
- RFC 2973 IS-IS Mesh Groups
- RFC 3373 Three-way Handshake for IS-IS Point-to-Point Adjacencies
- Draft-ietf-isis-restart-02 Restart Signaling for IS-IS
- Draft-ietf-isis-ipv6-06 Routing IPv6 with IS-IS
- Draft-ietf-isis-wg-multi-topology-11 Multi Topology (MT) Routing in IS-IS

QoS and VLAN Services

Quality of Service and Policies
- IEEE 802.1D – 1998 (802.1p) Packet Priority
- RFC 2474 DiffServ Precedence, including 8 queues/port
- RFC 2598 DiffServ Expedited Forwarding (EF)
- RFC 2597 DiffServ Assured Forwarding (AF)
- RFC 2475 DiffServ Core and Edge Router Functions

Traffic Engineering
- RFC 3784 IS-IS Externs for Traffic Engineering (wide metrics only)

VLAN Services: VLANs, vMANs
- IEEE 802.1Q VLAN Tagging
- IEEE 802.1v: VLAN classification by Protocol and Port

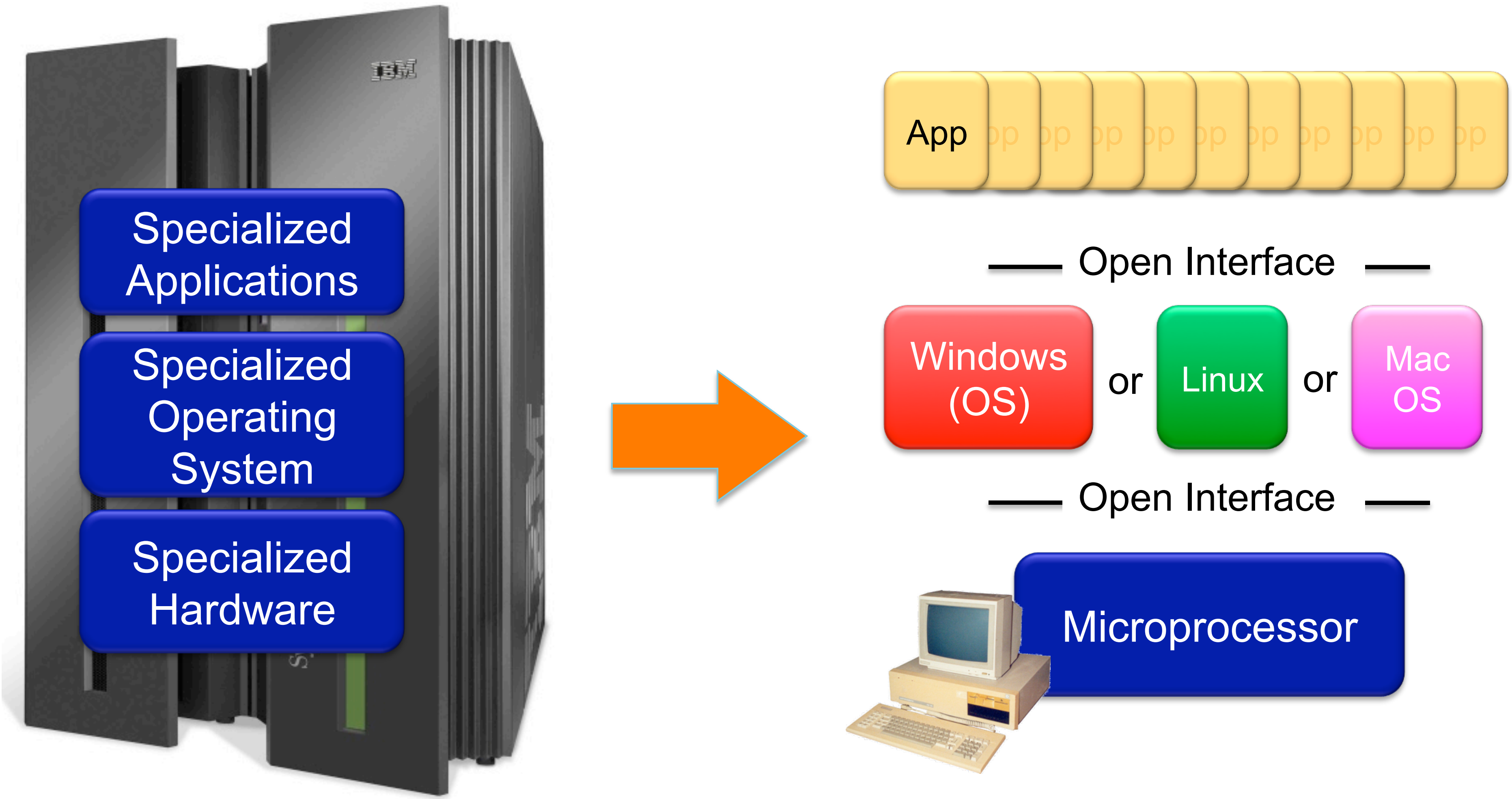
Advanced VLAN Services, MAC-in-MAC
- VLAN Translation in vMAN environments
- vMAN Translation
- IEEE 802.1ah/D1.2 Provider Backbone Bridges (PBB)/MAC-in-MAC

MPLS and VPN Services

Multi-Protocol Label Switching (MPLS)
Requires MPLS Layer 2 Feature Pack License
- RFC 2961 RSVP Refresh Overhead Reduction Extensions
- RFC 3031 Multiprotocol Label Switching Architecture
- RFC 3032 MPLS Label Stack Encoding
- RFC 3036 Label Distribution Protocol (LDP)
- RFC 3209 RSVP-TE: Extensions to RSVP for LSP Tunnels
- RFC 3630 Traffic Engineering Extensions to OSPFv2
- RFC 3784 IS-IS extensions for traffic engineering only (wide metrics only)
- RFC 3811 Definitions of Textual Conventions (TCs) for Multiprotocol Label Switching (MPLS) Management
- RFC 3812 Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Management Information Base (MIB)
- RFC 3813 Multiprotocol Label Switching (MPLS) Label Switching Router (LSR) Management Information Base (MIB)
- RFC 3815 Definitions of Managed Objects for the Multiprotocol Label Switching (MPLS), Label Distribution Protocol (LDP)
- RFC 4090 Fast Re-route Extensions to RSVP-TE for LSP (Detour Paths)
- RFC 4379 Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures (LSP Ping)
- draft-ietf-bfd-base-09.txt Bidirectional Forwarding Detection

Layer 2 VPNs
Requires MPLS Layer 2 Feature Pack License
- RFC 4447 Pseudowire Setup and Maintenance using the Label Distribution Protocol (LDP)
- RFC 4448 Encapsulation Methods for Transport of Ethernet over MPLS Networks
- RFC 4762 Virtual Private LAN Services (VPLS) using Label Distribution Protocol (LDP) Signaling
- RFC 5085 Pseudowire Virtual Circuit Connectivity Verification (VCCV)
- RFC 5542 Definitions of Textual Conventions for Pseudowire (PW) Management
- RFC 5601 Pseudowire (PW) Management

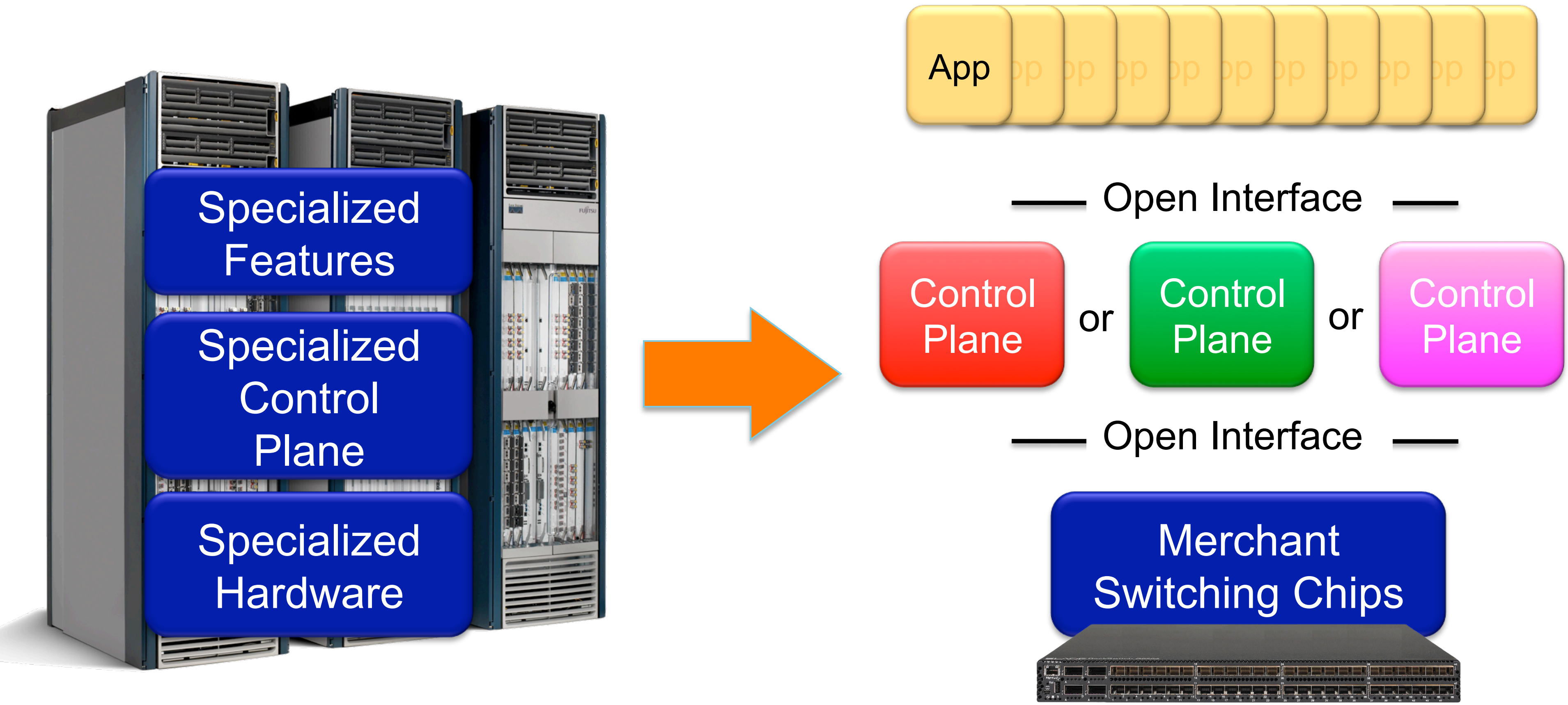
(slide by Nick McKeown, Stanford University)



Vertically integrated
Closed, proprietary
Slow innovation
Small industry

Horizontal
Open interfaces
Rapid innovation
Huge industry

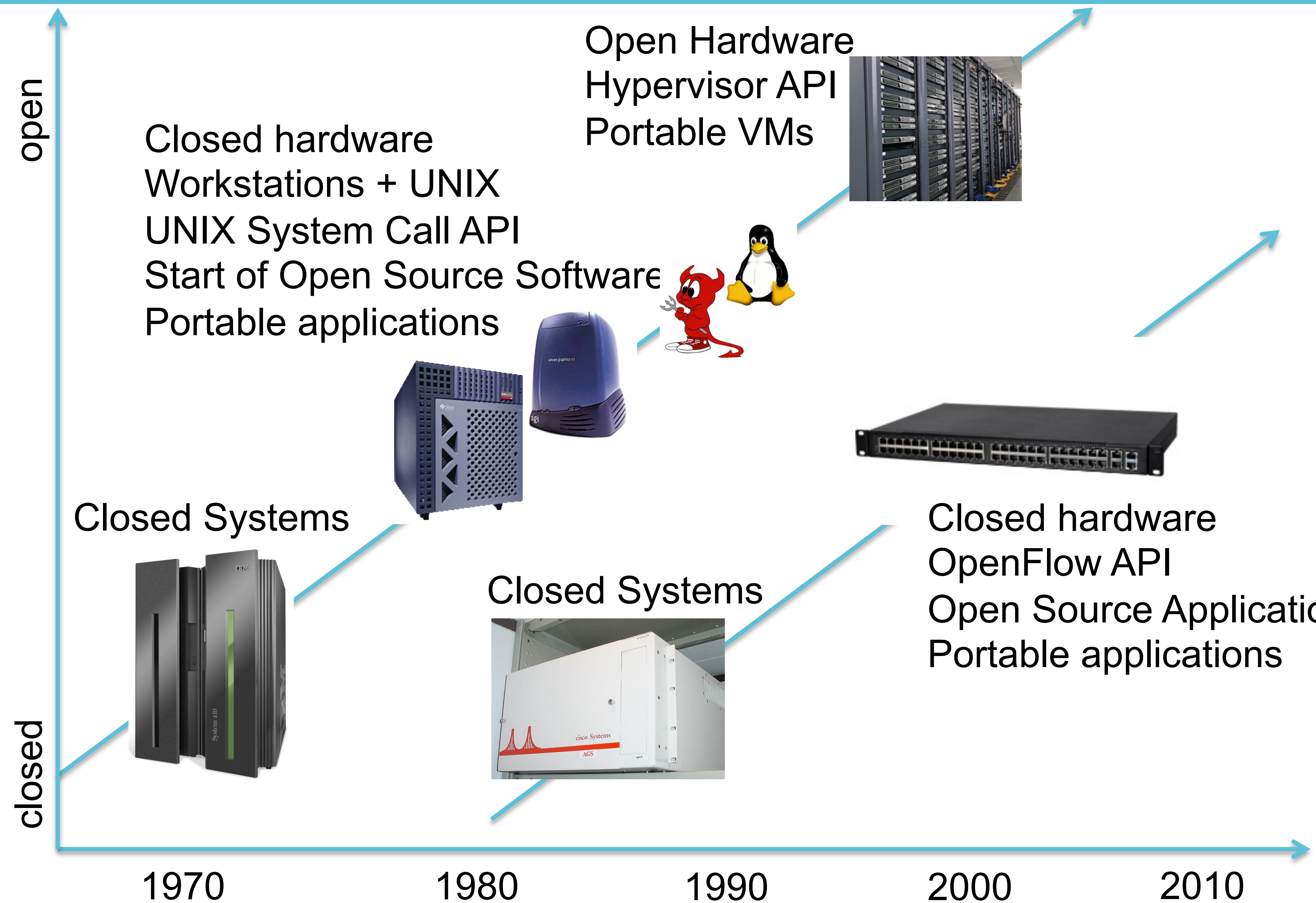
(slide by Nick McKeown, Stanford University)



Vertically integrated
Closed, proprietary
Slow innovation

Horizontal
Open interfaces
Rapid innovation

Computing vs Networking



Open Compute Project

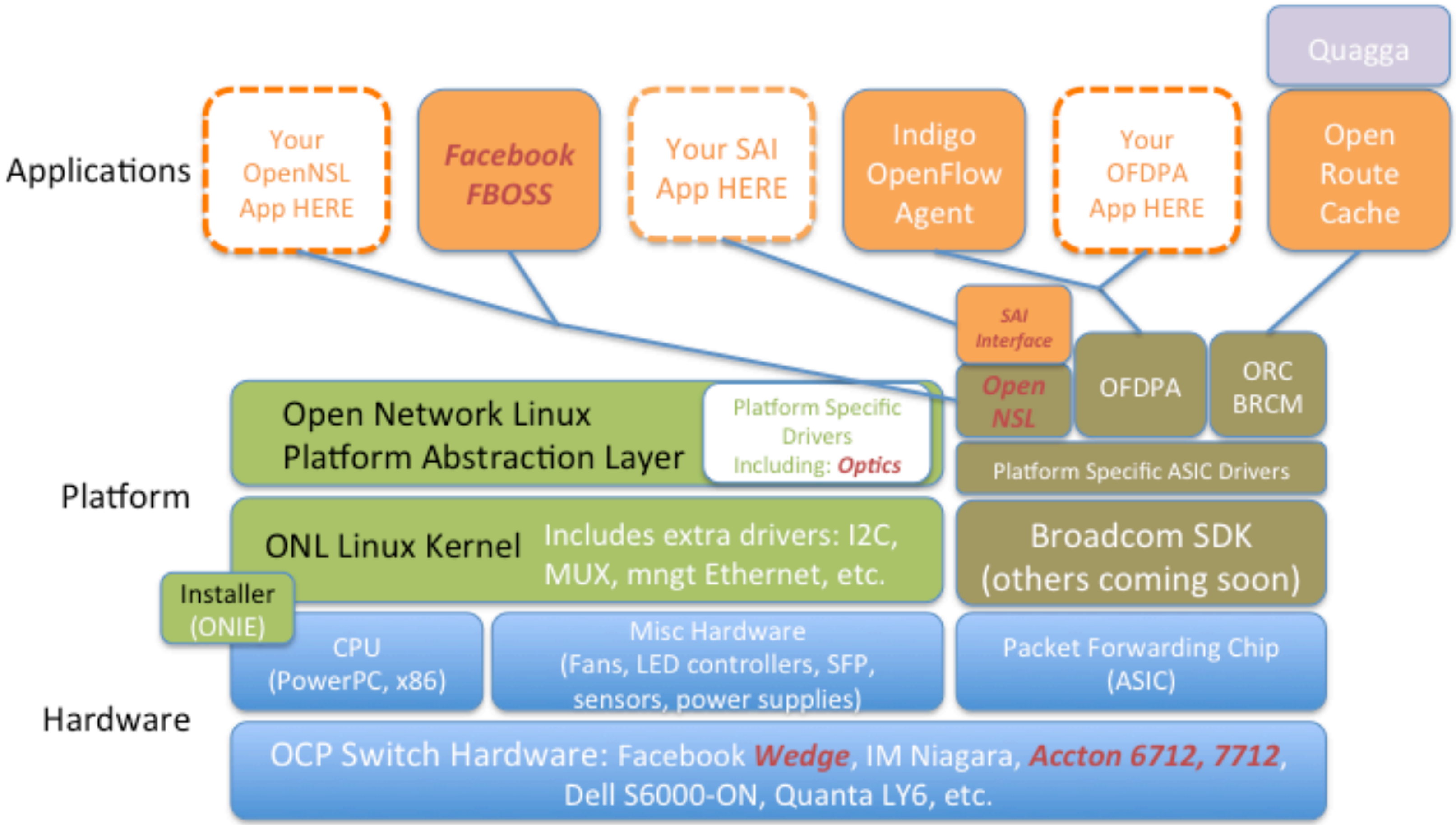
Started by Facebook in April 2011.
Share design of servers, data centres, etc. and collectively improve them.

Open Networking Project announced in May 2013.
Open design for a network switch.

Current Projects:

- Specs for open hardware switches (Accton/Edge-core, Facebook, Alpha, Broadcom/Interface Masters, Mellanox, Intel
- Specs for Switch Abstraction Interface (Microsoft, Dell, Facebook, Broadcom, Intel, Mellanox)
- Open Network Install Environment (Cumulus Networks)
- Open Network Linux (Big Switch Networks)

Emerging Open Switch Ecosystem



http://www.onie.org/

onie

ABOUT NEWS CONTRIBUTORS DOWNLOAD SUPPORT

Open Network Install Environment

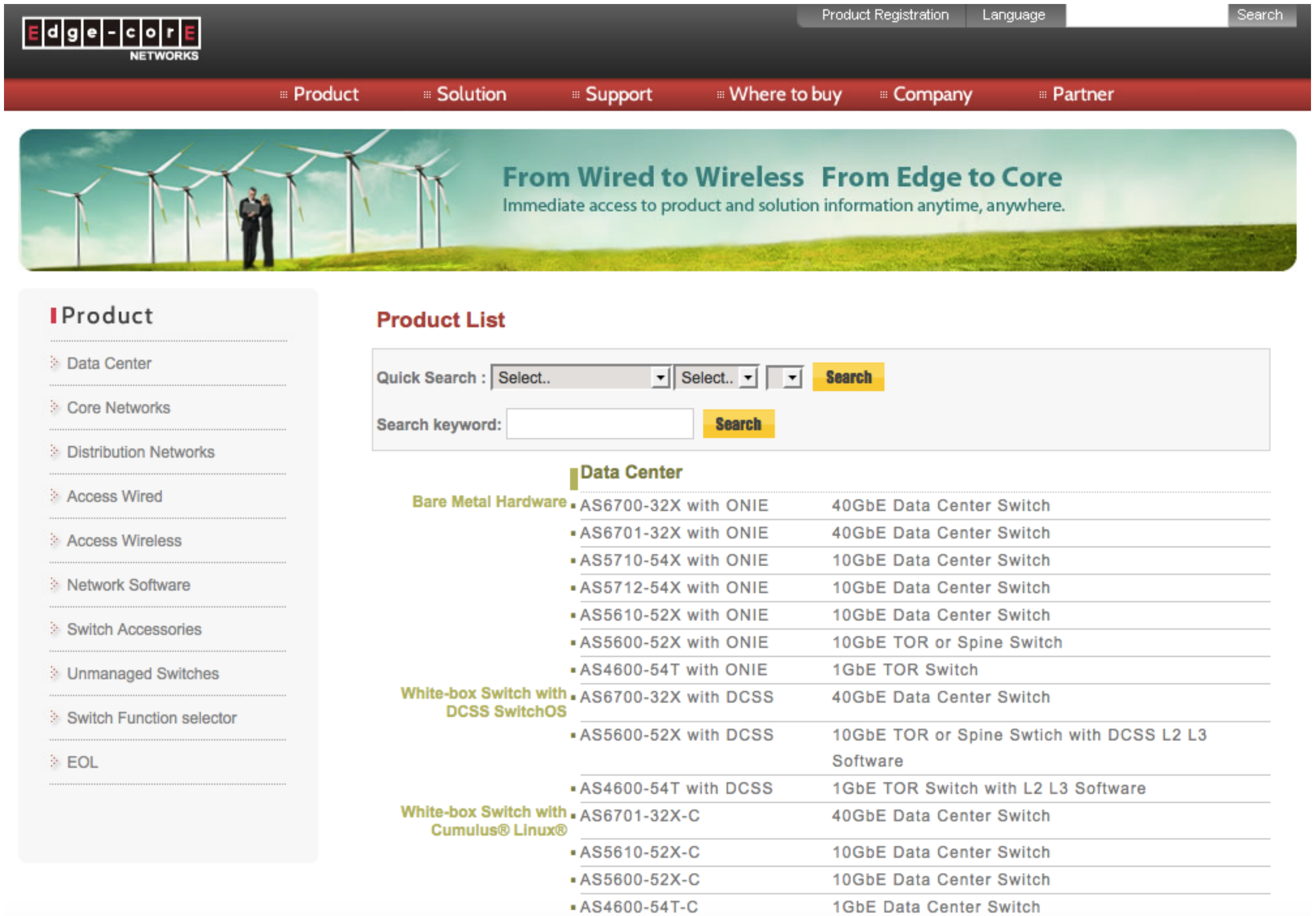
ONIE

ONIE is an Open Compute Project open source initiative contributed by Cumulus Networks that defines an open **"install environment"** for bare metal network switches

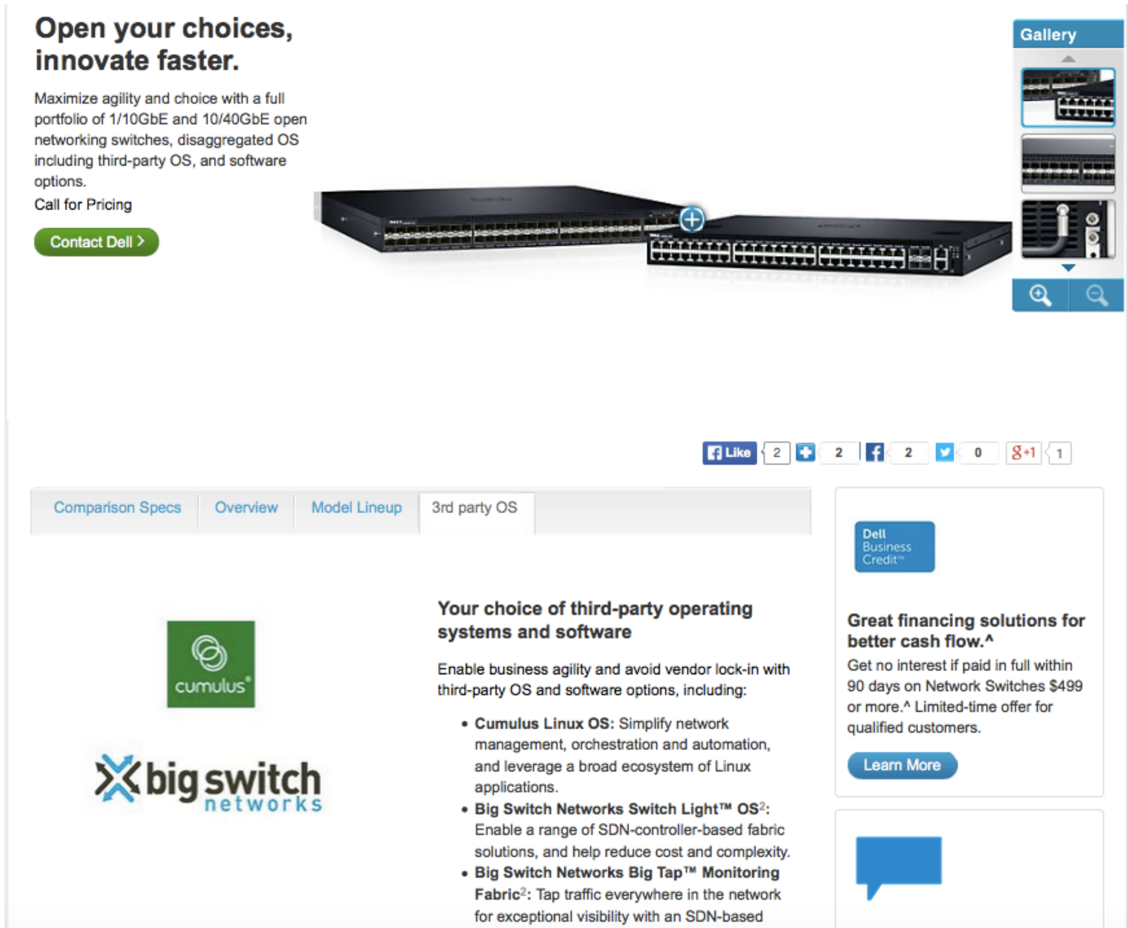
ABOUT ONIE DOWNLOAD

FaceBook Wedge 6-Pack

Edge-Core White Label Switches



Dell ONIE Switches



OpenNSL

Broadcom-Switch / **OpenNSL**

Watch 27 | Star 36 | Fork 4

branch: **master** | OpenNSL / **README.md**

simonknee on 9 Mar Initial commit

1 contributor

30 lines (21 sloc) | 0.82 kb — Raw | Blame | History

OpenNSL

OpenNSL Switch Specification and Software

Overview

Open Network Switch Library (OpenNSL) is a library of network switch APIs that is openly available for programming Broadcom network switch silicon based platforms. OpenNSL is a software interface with a set of open APIs that enable the development of new applications on top of Broadcom StrataXGS switches, giving customers the flexibility to tailor their network equipment and meet their unique infrastructure requirements.

Documentation

Detailed documentation can be found at `doc/html/index.html`.

OF-DPA

README.md

OF-DPA

OpenFlow 1.3.1 Switch Pipeline Specification and Software

Overview

Broadcom's OpenFlow Data Plane Abstraction (OF-DPA) is an application software component that implements an adaptation layer between OpenFlow and the Broadcom Silicon SDK. OF-DPA enables scalable implementation of OpenFlow 1.3 on Broadcom switch devices

Documentation

Detailed documentation can be found at `doc/html/index.html`.

http://www.opennetlinux.org/

Open Network Linux is a Linux distribution for "bare metal" switches, that is, network forwarding devices built from commodity components. ONL uses ONIE to install onto on-board flash memory. Open Network Linux is a part of the Open Compute Project and is a component in a growing collection of open source and commercial projects.

Software Defined Networking

Separation between Control Plane (policy) and Data Plane (packet forwarding).

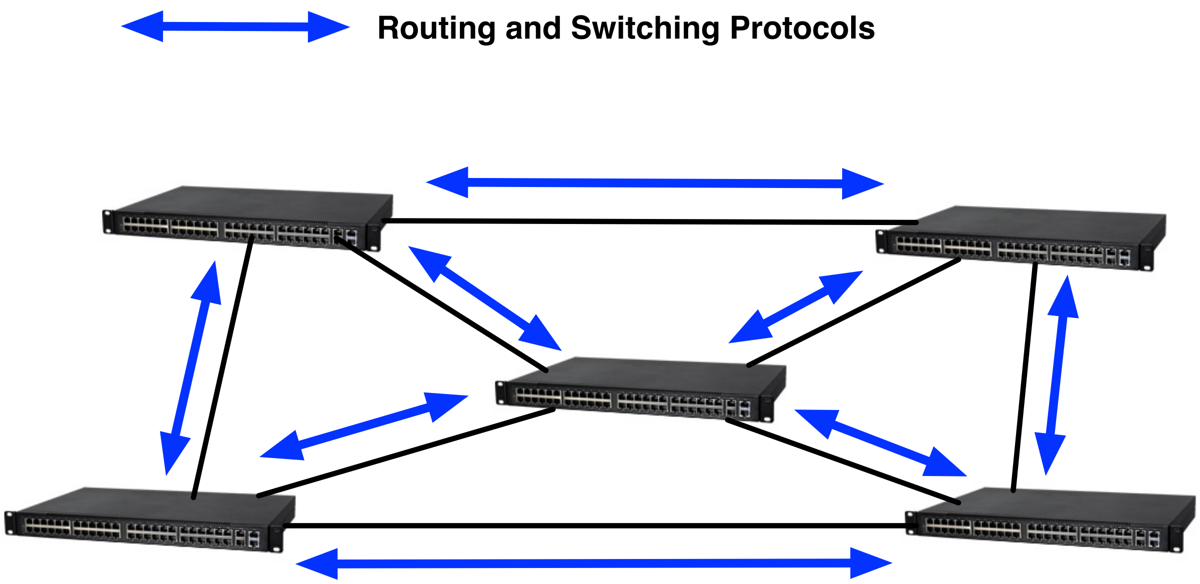
Logically centralised policy (control plane).
Program (controller) that reads policy configurations, compiles it to forwarding rules and sends those to network elements → Software Defined Networking.

“Dumb” switches (data plane).
Firmware does packet frame forwarding only.

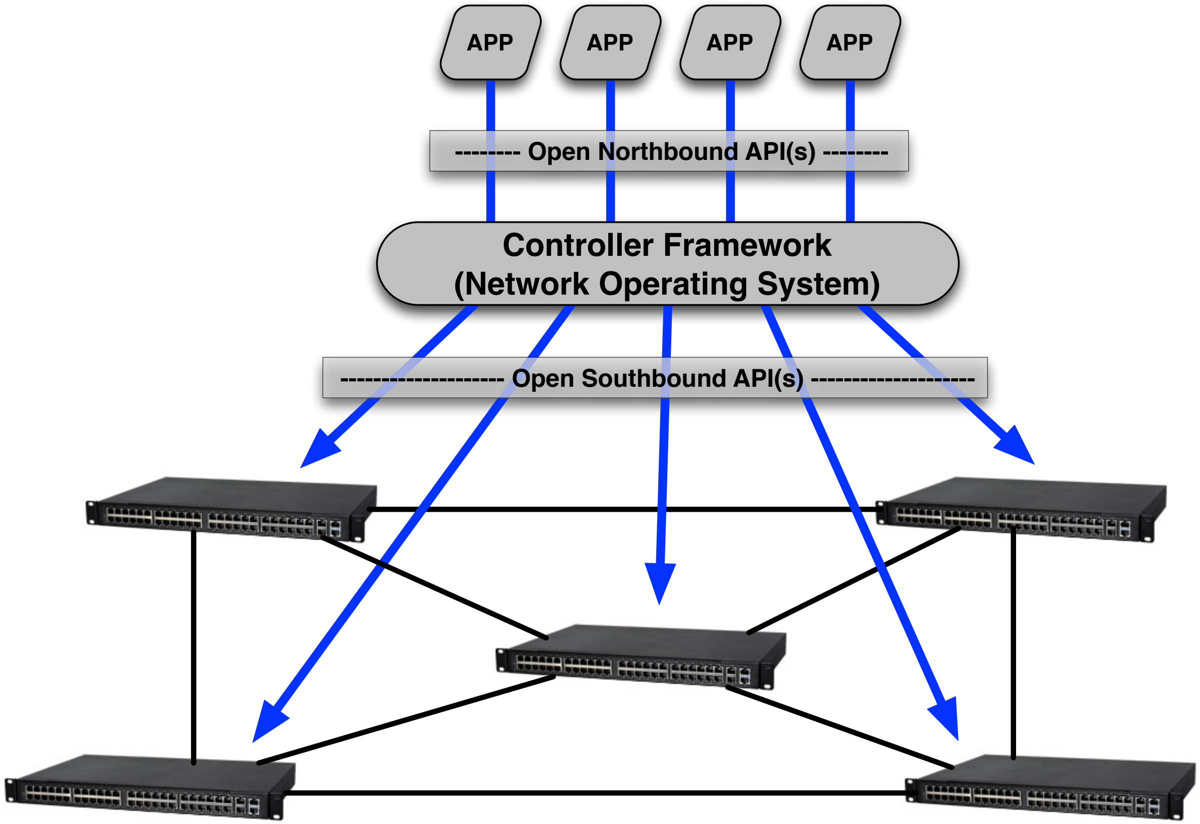
Standardised protocol between switches and controllers (e.g. OpenFlow).

Possibility of different vendors for switches and controllers.
More competition.

Traditional Networking



SDN Architecture



OpenFlow

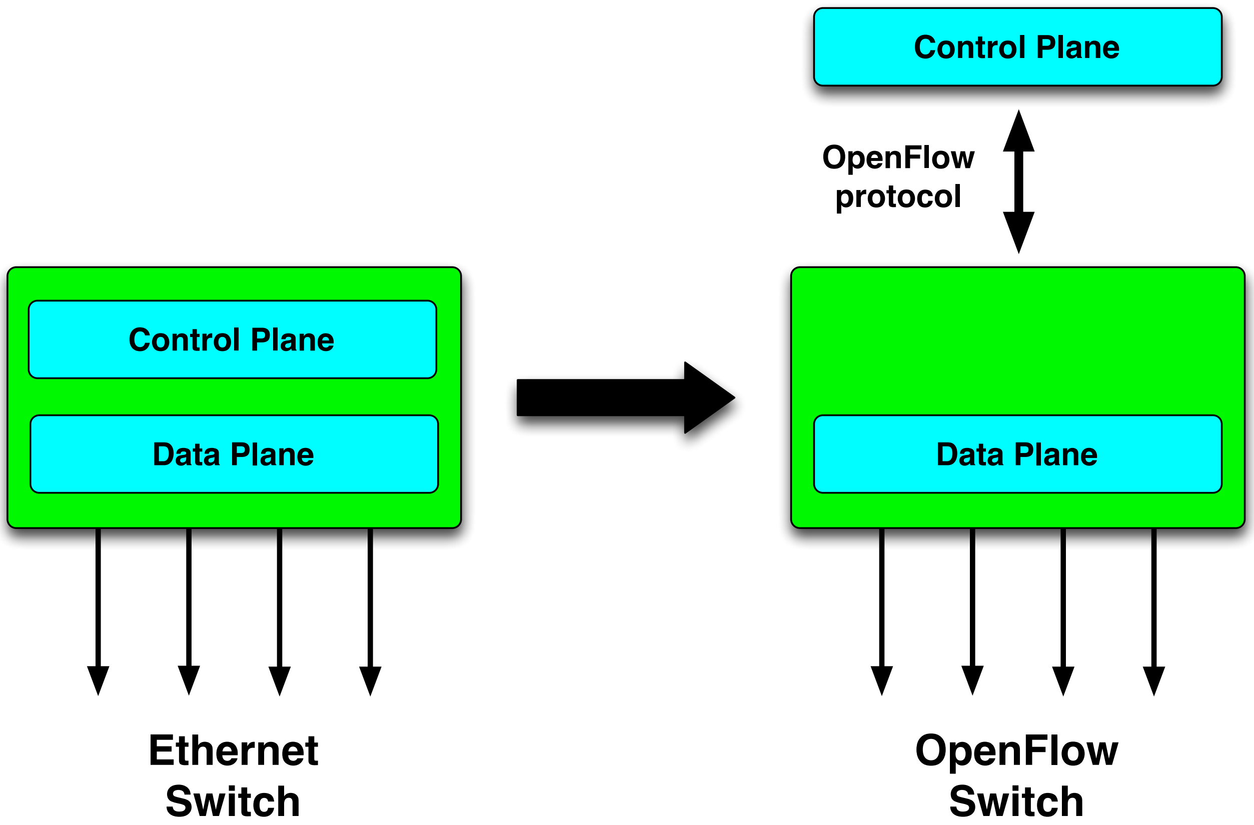
OpenFlow is low level language to manipulate switch forwarding table.

OpenFlow is the protocol between controller and switch.
Standardised protocol.

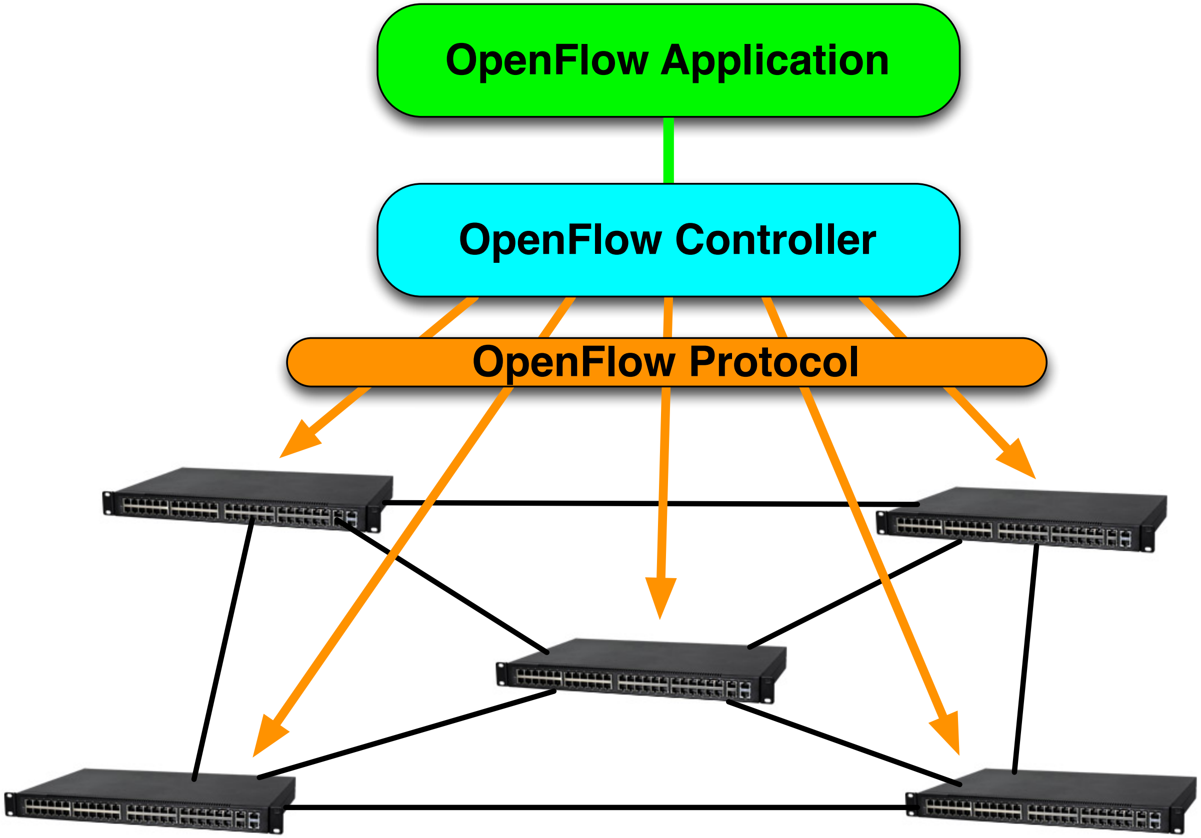
Many commercial available OpenFlow switches available.
Traditional Ethernet switches with an OpenFlow API.
Dedicated OpenFlow switches.

Many open source and proprietary OpenFlow controllers available.

Data and Control Plane Separation



OpenFlow Controlled Network



OpenFlow Standardisation

OpenFlow is standardised by the Open Networking Foundation (ONF).

ONF is a non-profit consortium.

Founded in March 2011 by Deutsche Telecom, Facebook, Google, Microsoft, Verizon and Yahoo!

Most vendors in ICT and networking are members now.

Mission:
The Open Networking Foundation (ONF) is a user-driven organization dedicated to the promotion and adoption of Software Defined Networking (SDN) through open standards development.

OpenFlow Protocol Standards

OpenFlow 1.0.0 (December 2009)
Most widely used version
OpenFlow 1.1.0 (February 2011)
OpenFlow 1.2 (December 2011)
IPv6 support, extensible matches
OpenFlow 1.3.0 (June 2012)
Flexible table miss, per flow meters, PBB support
OpenFlow 1.4.0 (October 2013)

OF-Config 1.0 (December 2011)
OF-Config 1.1 (January 2012)
OF-Config 1.2 (2014)

OpenFlow Test
Interoperability Event technical papers

OpenFlow Protocol

Insert flow forwarding entries in switches.

Send packets to OpenFlow switch data path.

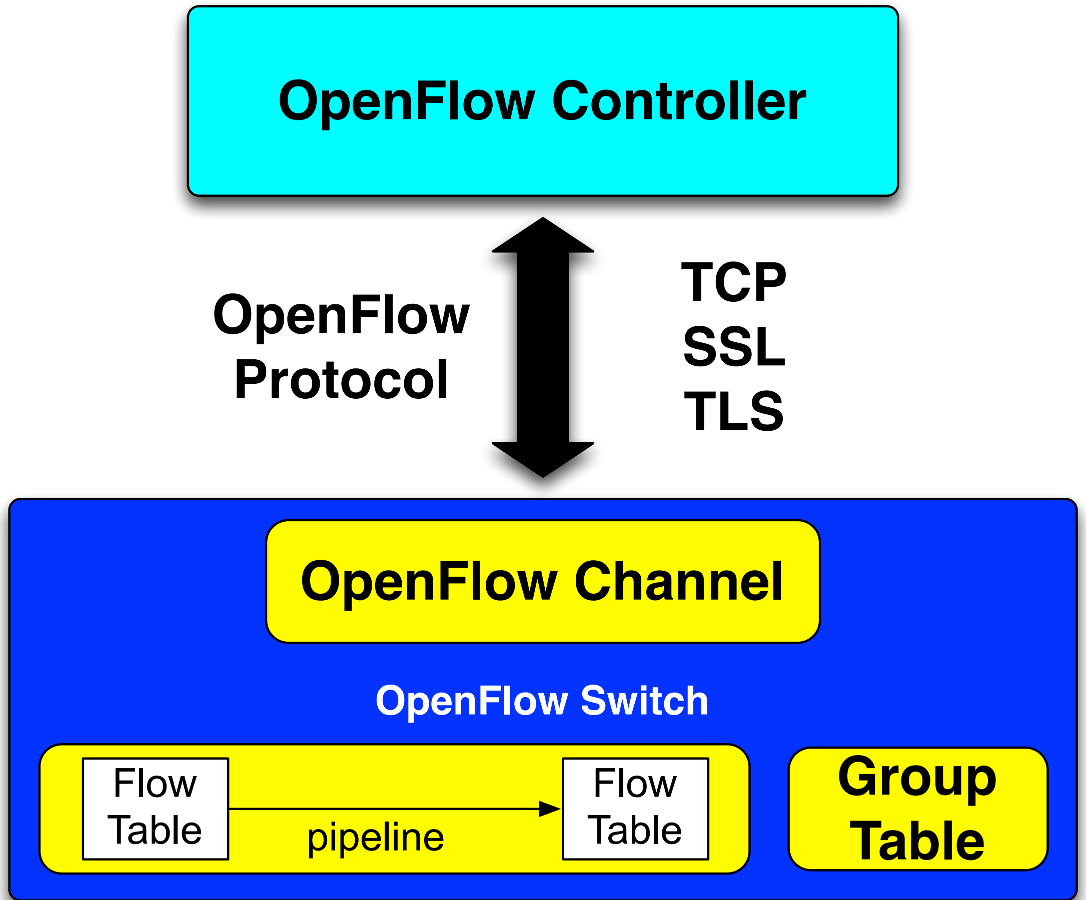
Receive packets from OpenFlow switch data path.

Retrieve data path traffic statistics from OpenFlow switch.

Retrieve flow tables from OpenFlow switch.

Retrieve parameters from OpenFlow switch.
E.g. number and properties of ports.

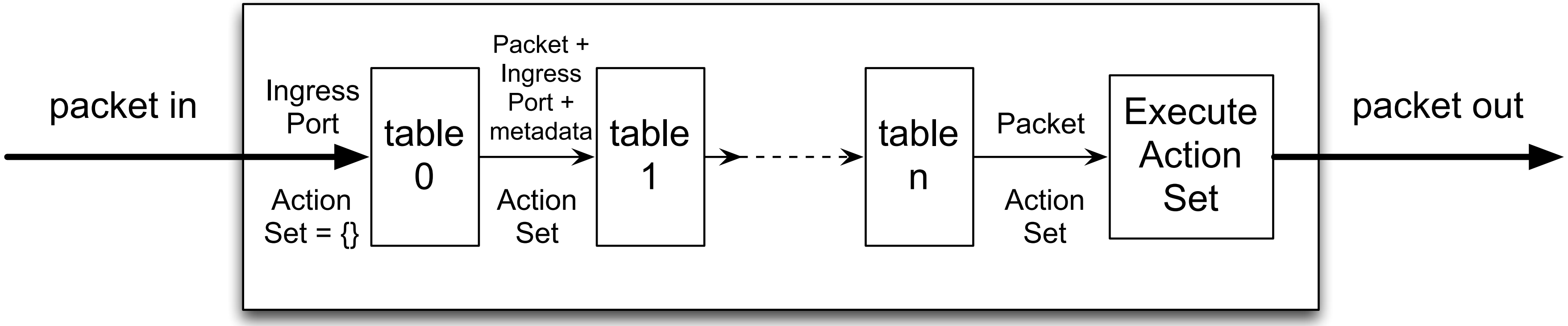
OpenFlow Components



Flow Table

Matching rule #1	Counter	Action #1
Matching rule #2	Counter	Action #2
Matching rule #3	Counter	Action #3
Matching rule #4	Counter	Action #4
Matching rule #5	Counter	Action #5
Matching rule #6	Counter	Action #6
Matching rule #7	Counter	Action #7

Table Pipeline



Header Matching (OF 1.3)

Input port
Metadata passed between tables
Ethernet source/destination address
Ethernet type
VLAN ID
VLAN priority
IP DSCP (6 bits in ToS field)
IP ECN (2 bits in ToS field)
IP protocol
IPv4/IPv6 source/destination address
TCP/UDP/SCTP ***source/destination port***
ICMP/ICMPv6 type/code
ARP opcode
ARP src/tgt IPv4/hardware address
IPv6 flow label, extension header
ND target address
ND src/tgt link layer address
MPLS label, traffic class, bottom of stack bit
PBB I-SID
Logical port metadata

Actions

Output *port_nr*
Group *group_id*
Drop
Set-Queue *queue_id*
Push-Tag/Pop-Tag *ethertype*
Set-Field *field_type value*
Change-TTL *ttl*

Group Table

Identifier	**Group Type**	**Counters**	**Action Buckets**

Group Types

Indirect: Execute the single bucket in this group

Usage: multiple flow entries can point to this group ID, bucket action can be IP routing next hop

ALL: Execute all buckets

Used for multicast and broadcast

Select: Execute one bucket in the group

Used for load balancing

Fast Failover: Execute the first live bucket

Each action bucket is associated with a port

Flow Insertion

Proactive

Flow entries are inserted in the OpenFlow switches before packets arrive

Reactive

Packets arriving at an OpenFlow switch without a matching flow entry are sent to OpenFlow controller. They examined by the controller after which flow entries are inserted in the switches

Example of Proactive Flow Entries

Forward all packets between port 1 and 2

```
ovs-ofctl add-flow br0 in_port=1,actions=output:2
ovs-ofctl add-flow br0 in_port=2,actions=output:1
```

Forward all packets between access port 4 and trunk port 6 using VLAN ID 42

```
ovs-ofctl add-flow br0 in_port=4,
actions=push_vlan:0x8100,set_field:42->vlan_vid,output:6

ovs-ofctl add-flow br0 in_port=6,
actions=strip_vlan,output:4
```

Open Networking Research Center

Located at Stanford University & UC Berkeley

Sponsors: CableLabs, Cisco, Ericsson, Google, Hewlett Packard, Huawei, Intel, Juniper, NEC, NTT Docomo, Texas Instruments, Vmware

People:
Nick McKeown @ Stanford University
Scott Shenker @ UC Berkeley

http://onrc.stanford.edu/

ON.LAB

Headed by Guru Parulkar

Professor at Stanford University

Build open source OpenFlow tools and platforms

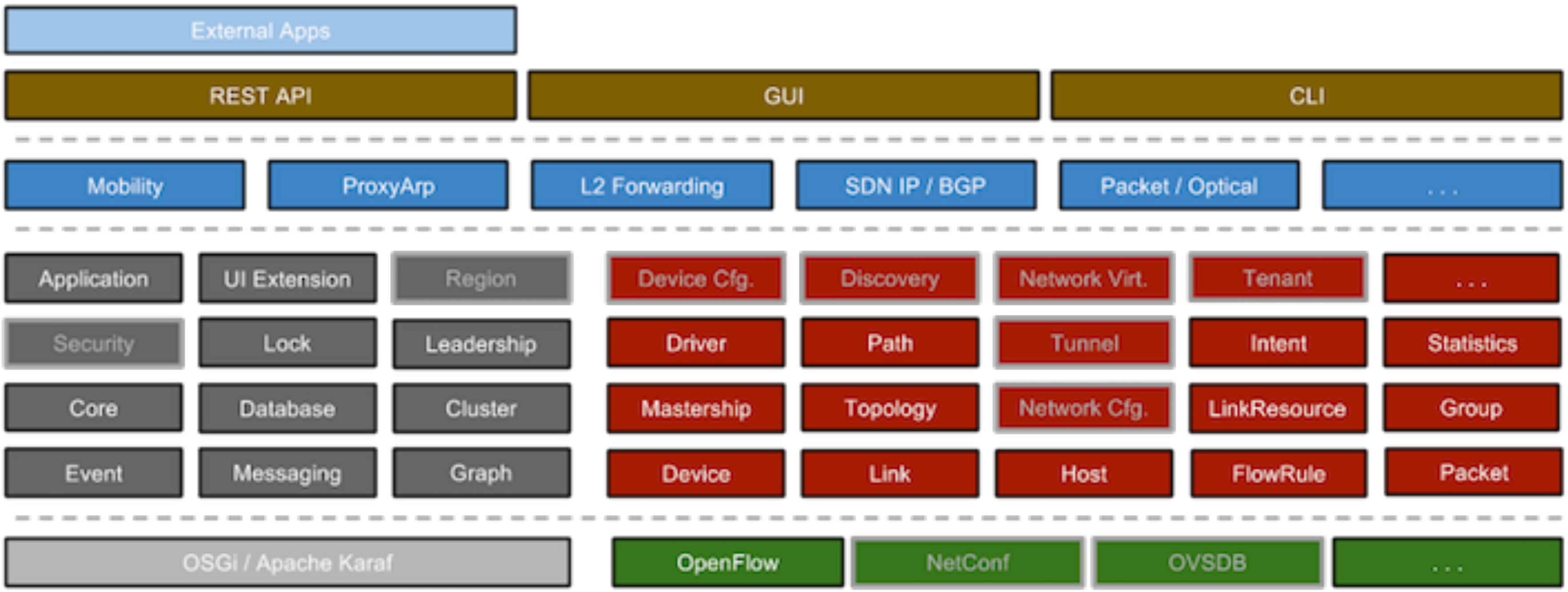
Beacon, NOX, FlowVisor, Mininet

http://onlab.us/

ONOS Members

Partners

ONOS



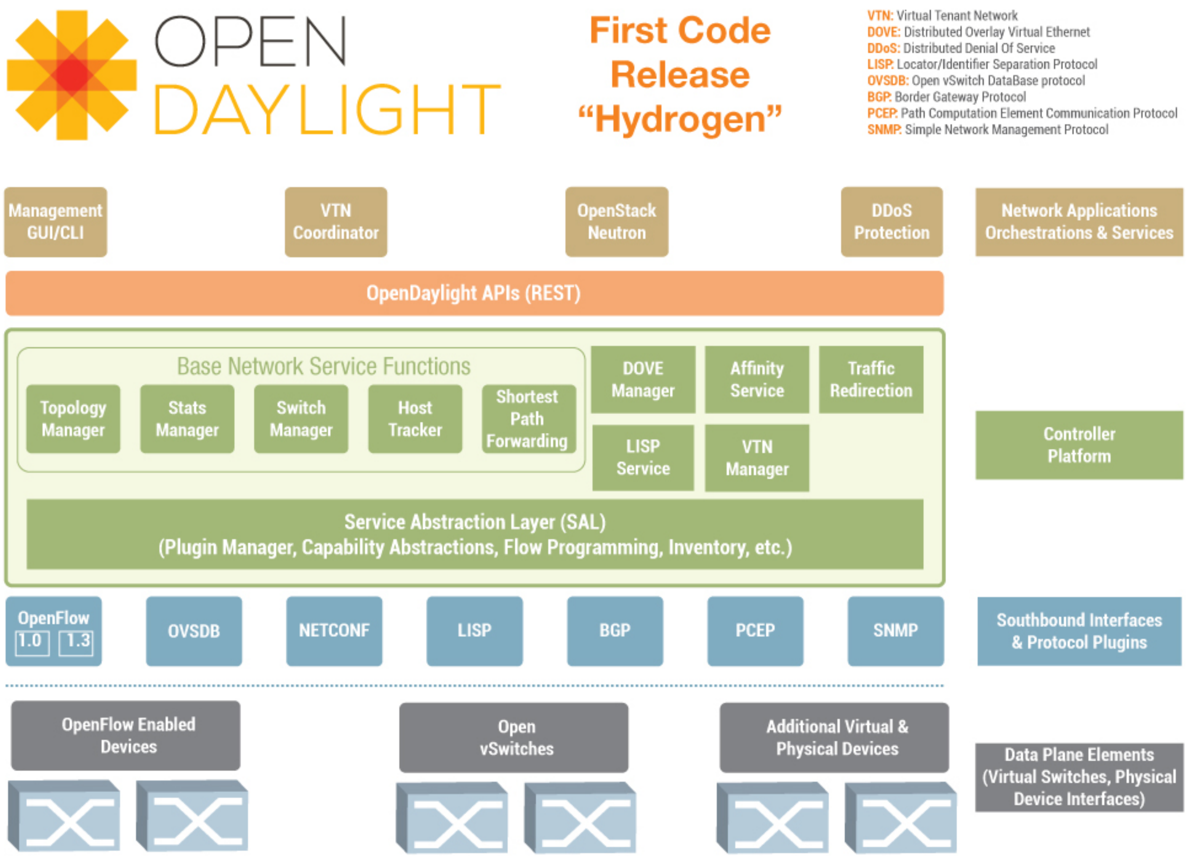
External Apps
REST API
GUI
CLI
Mobility
ProxyArp
L2 Forwarding
SDN IP / BGP
Packet / Optical
. . .
Application
UI Extension
Region
Security
Lock
Leadership
Core
Database
Cluster
Event
Messaging
Graph
Device Cfg.
Discovery
Network Virt.
Tenant
. . .
Driver
Path
Tunnel
Intent
Statistics
Mastership
Topology
Network Cfg.
LinkResource
Group
Device
Link
Host
FlowRule
Packet
OSGi / Apache Karaf
OpenFlow
NetConf
OVSDB
. . .

OpenDaylight Members

PLATINUM MEMBERS

BROCADE
CISCO
CITRIX
DELL
ERICSSON
hp
IBM
intel
Microsoft
redhat

OpenDaylight Architecture



Google Data Network

Google has two networks:

I-Scale: User facing services (search, YouTube, Gmail, etc), high SLA
G-Scale: Data centre traffic (intra and inter), lower SLA, perfect for OpenFlow testing

OpenFlow introduced in G-Scale network since mid 2010

Experience/benefits of introducing OpenFlow:

Better Traffic Engineering (global view of network)

Centralised Traffic Engineering much faster on a 32 core server (25-50 times as fast) than on slow CPUs inside switches

Software development for a high performance server with modern software tools (debuggers, etc) much easier and faster and produces higher quality software than development for an embedded system (router/switch) with slow CPU and little memory

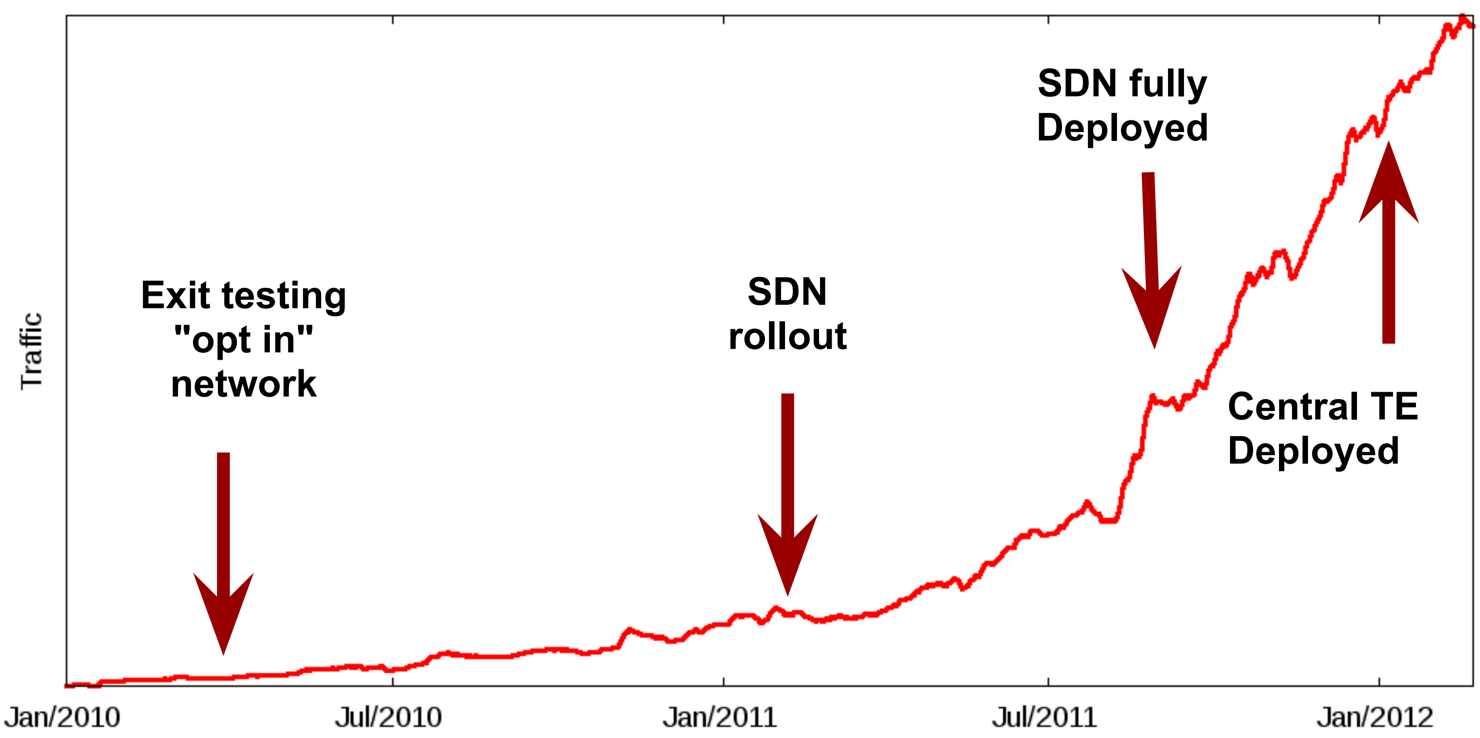
Google Data Network

Google OpenFlow Switch (source Google)

- Built from merchant silicon
  - 100s of ports of nonblocking 10GE
- OpenFlow support
- Open source routing stacks for BGP, ISIS
- Does not have all features
  - No support for AppleTalk...
- Multiple chassis per site
  - Fault tolerance
  - Scale to multiple Tbps

Google's OpenFlow Deployment

G-Scale WAN Usage

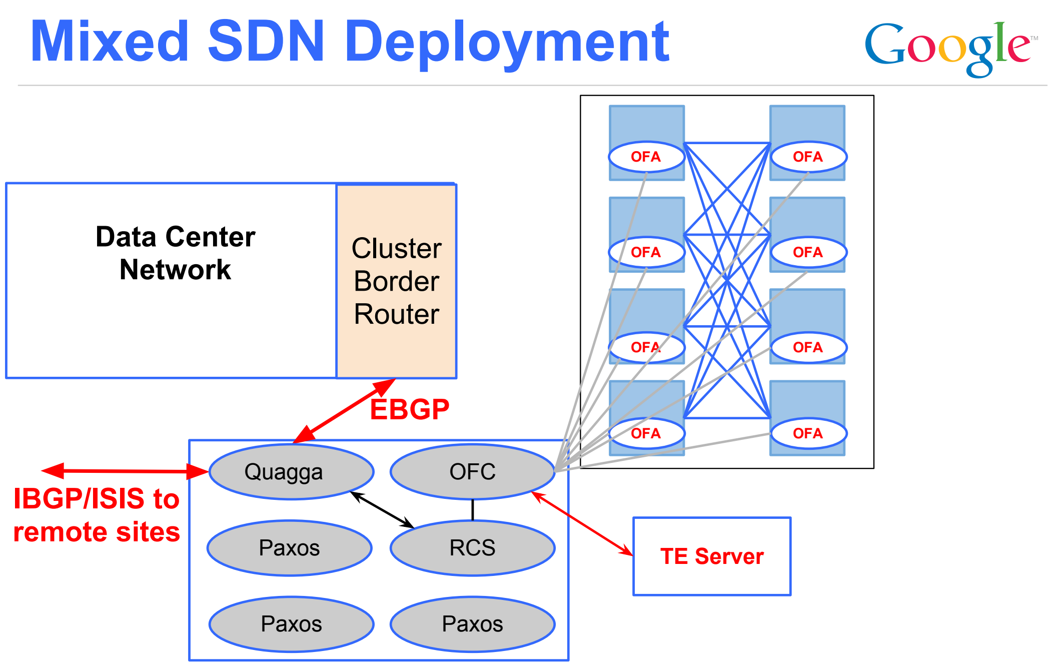


Google Data Network

Multiple controllers.
3, 5, 7 with Paxos election system.

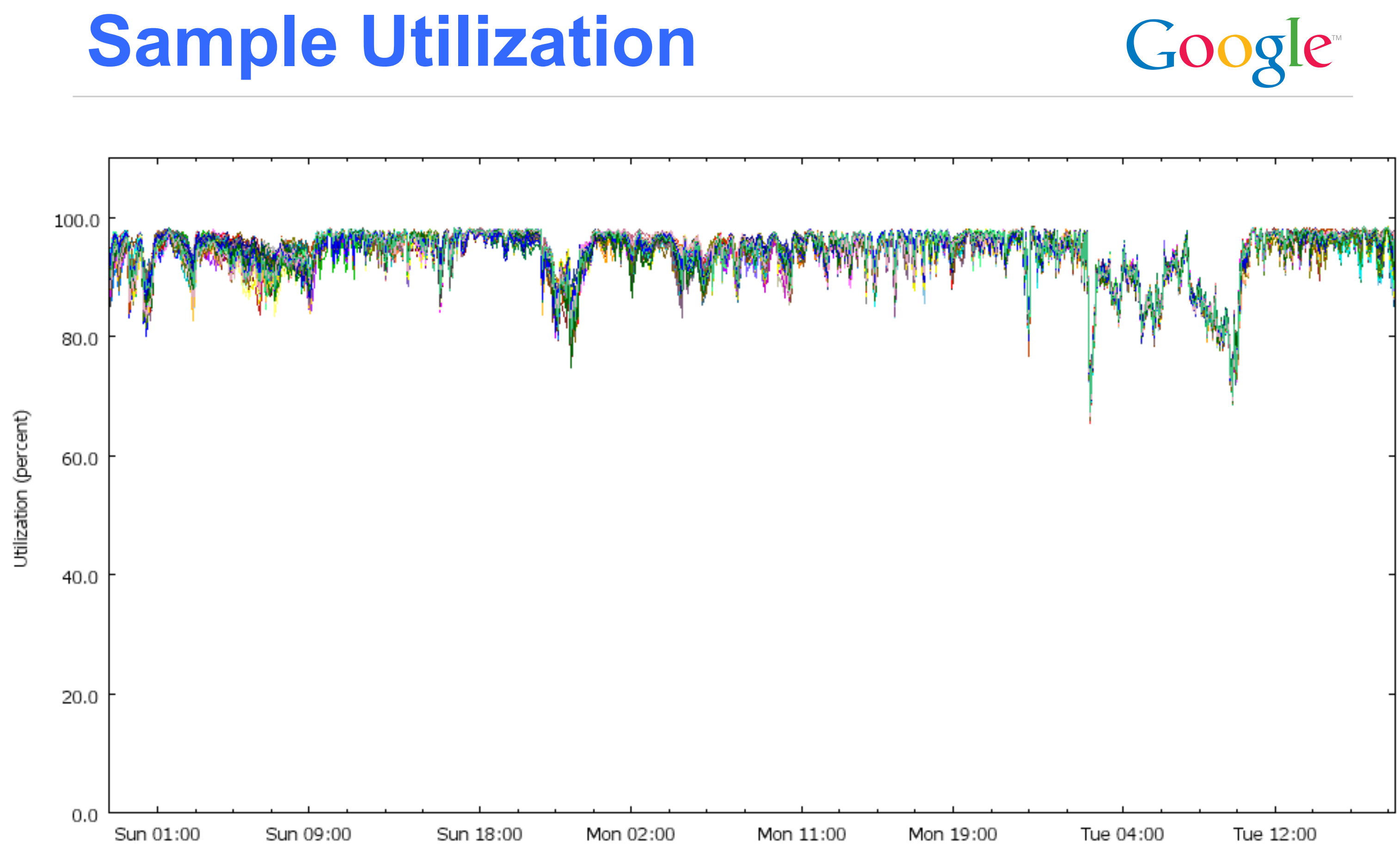
The whole network is emulated in a simulator.
New software revisions can be tested in the simulator.
Network events (e.g. link down) are sent to production servers + testbed.
Testing in simulator but with real network events.

Google OpenFlow Architecture



- Ready to introduce new functionality, e.g., TE

Almost 100% Link Utilization



Google Data Network

Experience/benefits:

Software development for a high performance server with modern software tools (debuggers, etc) much easier and faster and produces higher quality software than development for an embedded system (router/switch) with slow CPU and little memory.

Centralised Traffic Engineering much faster on a 32 core server (25-50 times as fast).

OpenFlow Network Service Prototype

GN3plus Open Call Project (CoCo).

October 2013 – March 2015 (18 months).

Budget Eur 216K.

16.4 person months.

Partners: SURFnet (NL) & TNO (NL).

Five work packages:
WP1: use cases & market demand
WP2: architecture, design & development
WP3: experimental validation
WP4: dissimination
WP5: project management

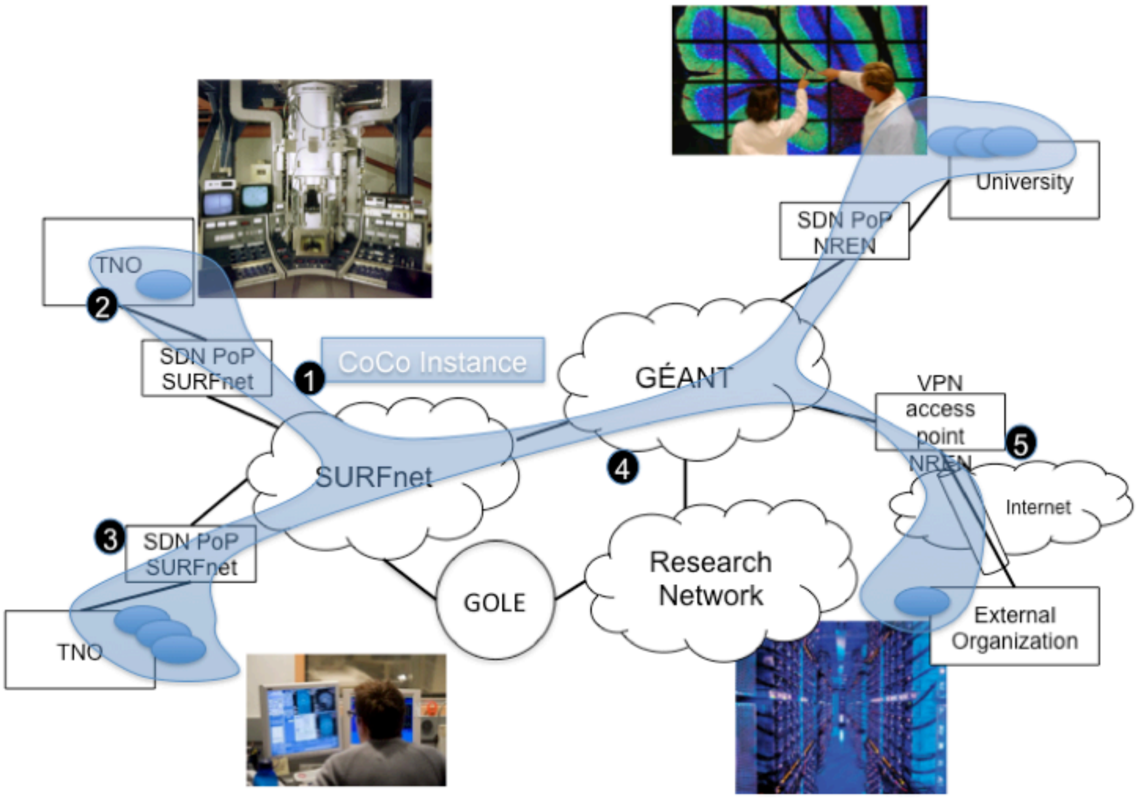
Community Connection (CoCo) Service

Goal of CoCo service:

On-demand private multi-domain, multi-point networks.
Connect laptops, VMs, storage, instruments, eScience resources.
Each eScience community group can easily setup their own private CoCo instance via web portal.

Based on OpenFlow programmable network infrastructure.

Example CoCo Instance



Forwarding in CoCo

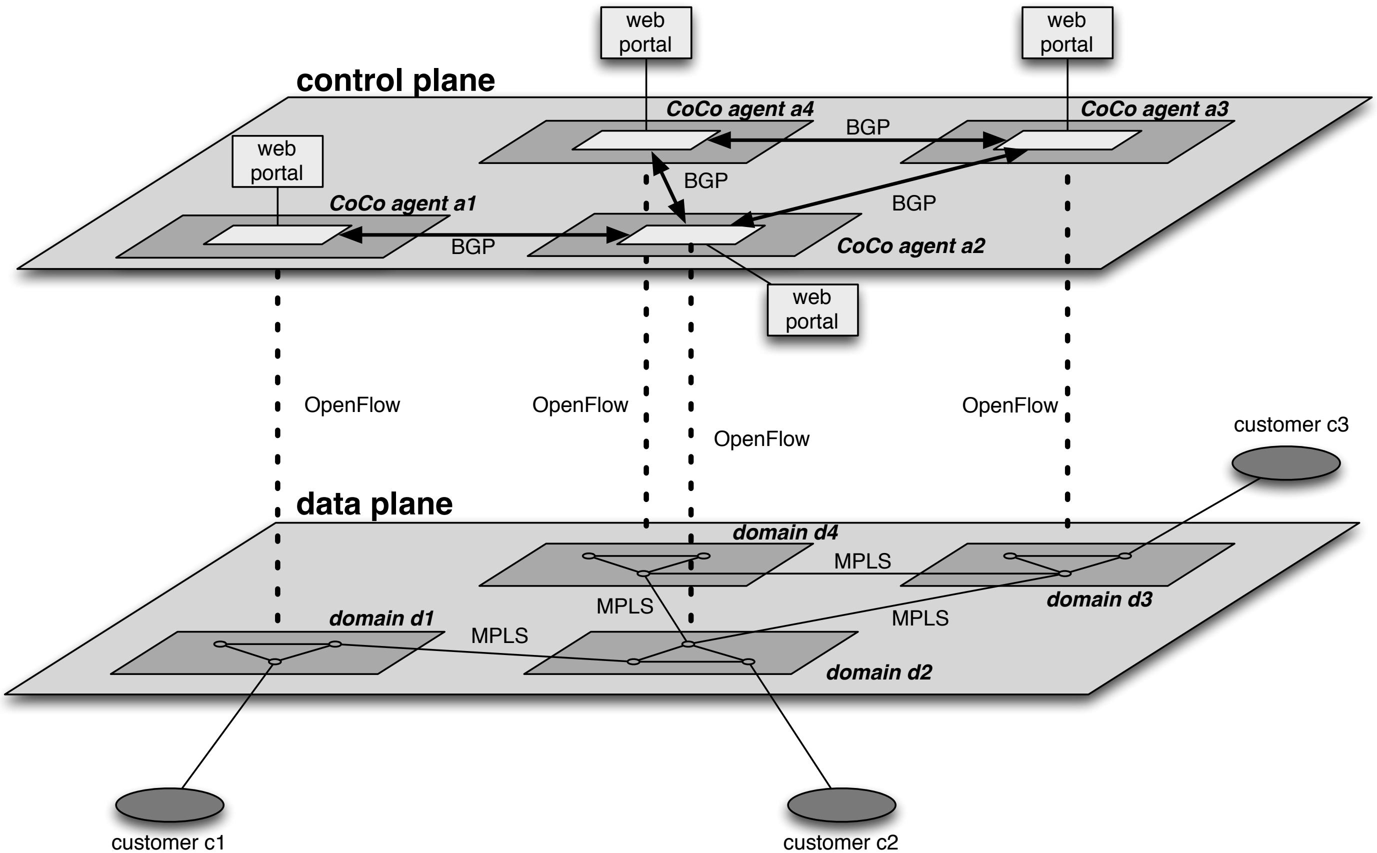
CoCo OpenFlow application inserts flow forwarding entries.

Forwarding based on MPLS label matching and forwarding.

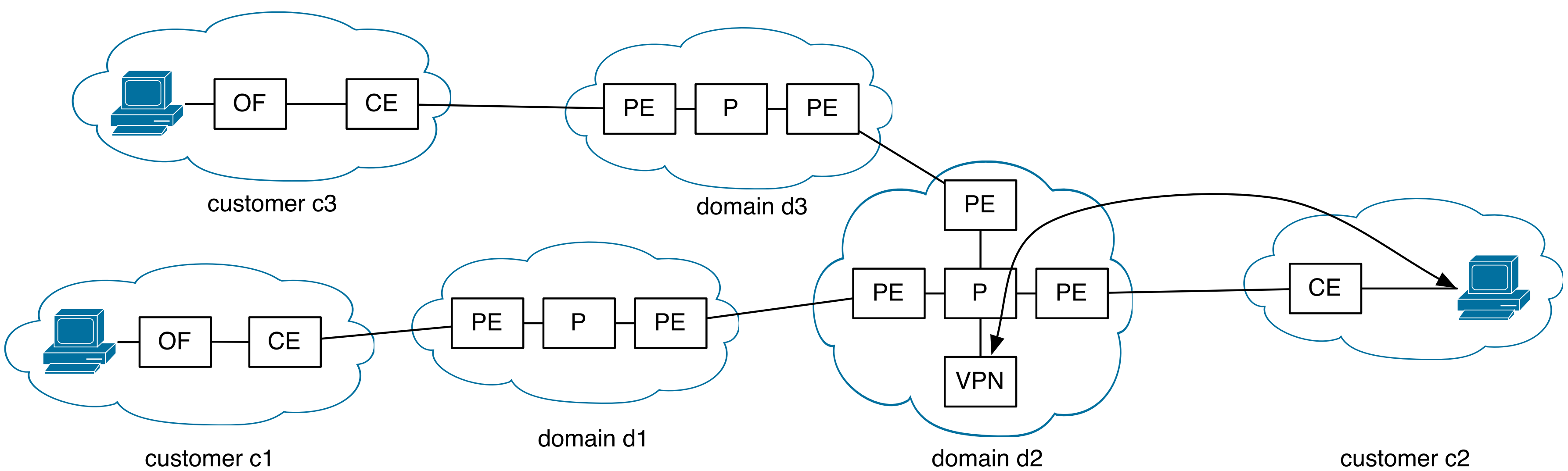
Label stack with two MPLS labels.
Outer MPLS label used to identify egress PE switch.
Inner MPLS label used to identify CoCo instance.

Adding and removing MPLS labels done at edges (PE).

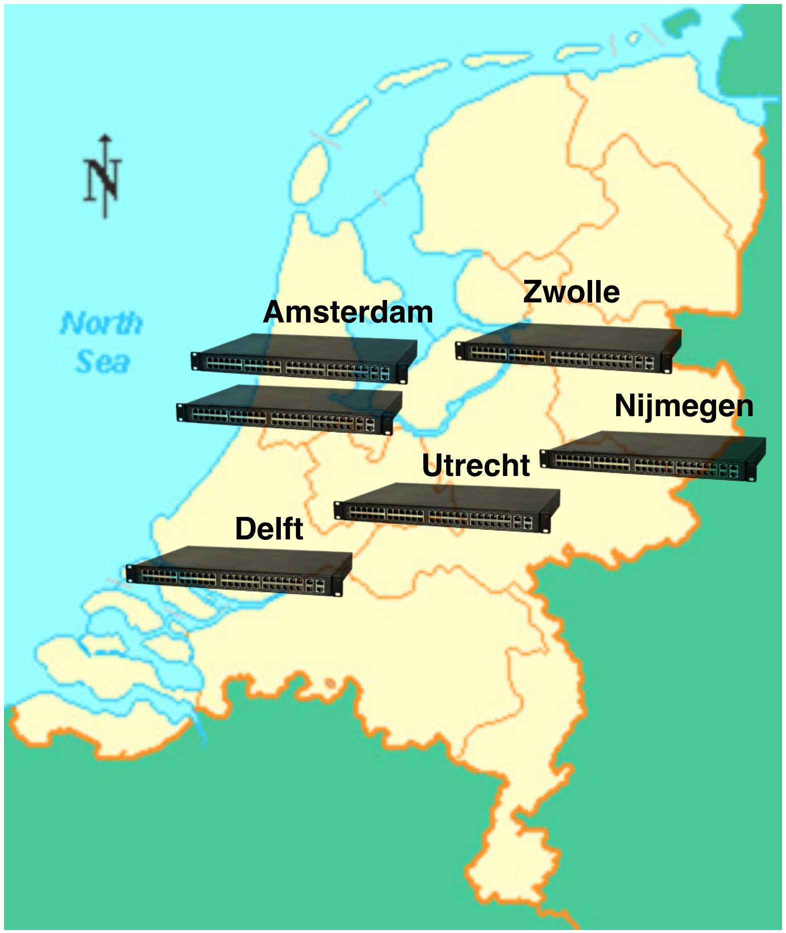
CoCo Inter-Domain Architecture



CoCo Inter-Domain Forwarding



SURFnet OpenFlow Testbed



- 5 sites
- Co-located at SURFnet7 core sites
- 1GE overlay over SURFnet7
- Looped multi-stage
- 4 redundant paths between each pair of switches
- Each site co-located with a small OpenStack cloud
- OpenDaylight controller
- Initial OF application will offer functionality for L3-VPN and L2 P2P

Open vSwitch

Software switch that implements the OpenFlow protocol

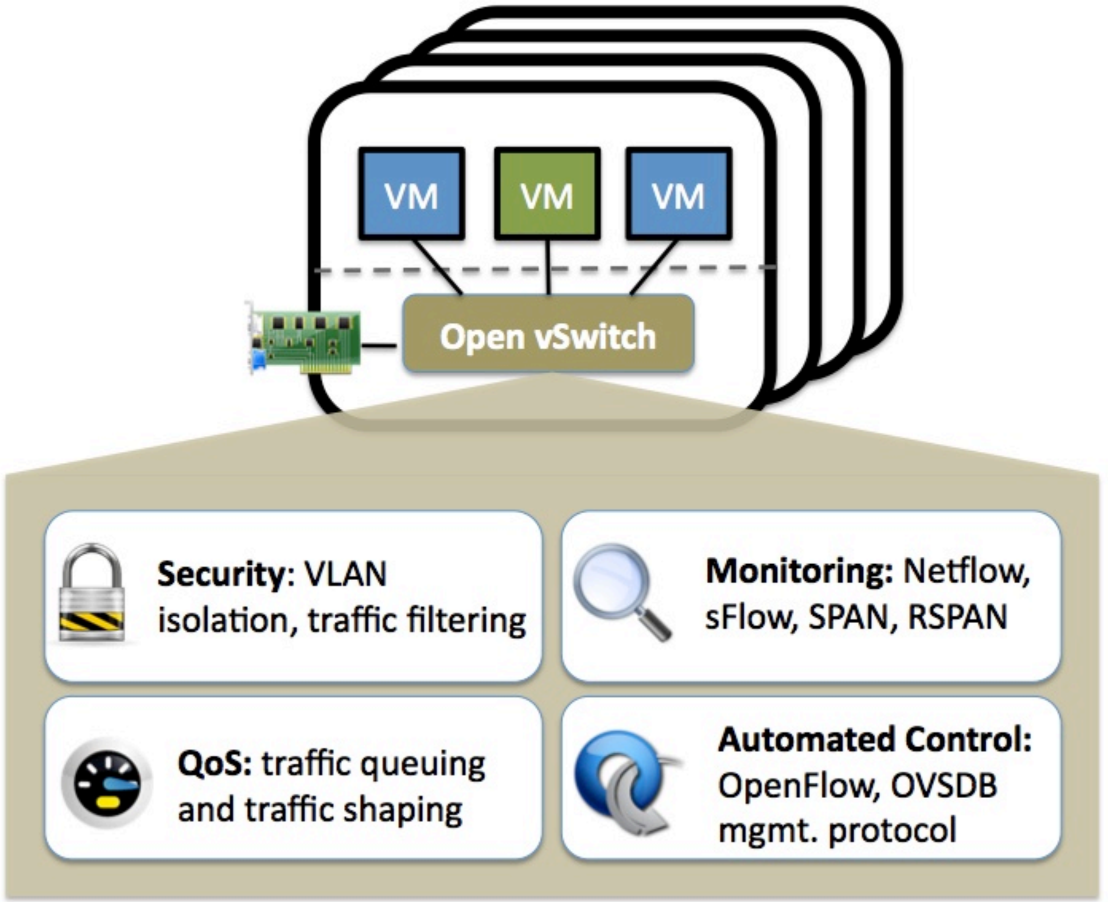
- Open Source project
- Included in the Linux kernel, OpenStack, OpenNebula, …

Developed by Nicira (startup founded in 2007)

- Martin Casado (Stanford University)
- Nick McKeown (Stanford University)
- Scott Shenker (UC Berkeley)

Nicira was acquired by VMware in 2012 for USD 1.26 billion

Open vSwitch in a Cloud Environment



Network Functions Virtualisation (NFV)

ETSI Industry Specification Group.

Goal:
Provide Network Functions through virtualisation techniques using general purpose servers and storage devices.

How:
Replace proprietary hardware network appliances by consolidating the network functions as applications running on virtual machines.

Summary

Networking moving to open hardware and open APIs

Introduction of abstractions and hiding of complexity

New companies focussing on one part of this new ecosystem

Software is playing an increasingly important role in networking

Ronal van der Pol
Ronald.vanderPol@SURFnet.nl

WHAT SURF CAN DO