What Happened To OpenFlow?

SNE GUEST LECTURE 12 MAY 2017

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

SURF NET

Outline

Concepts behind OpenFlow

- Early days at Stanford University

Software Defined Networking

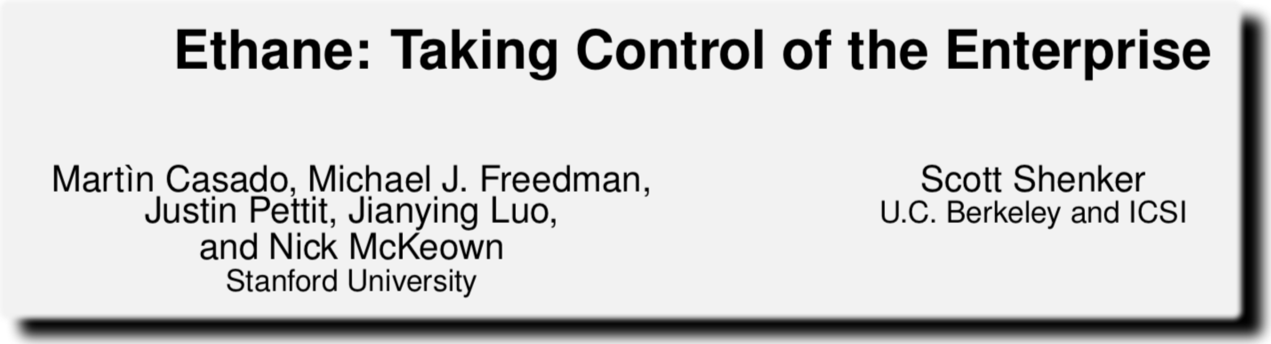
- SDN as a meaningless acronym
- The hyperscales (Facebook, Google, Facebook)
- Network Disaggregation

What are we working on at SURFnet

- Smart Edge, containers and serverless applications (eBPF, P4)
- Network Disaggregation (SnapRoute, SONiC, FRR)

P4 demo

Stanford University Papers 2007/2008

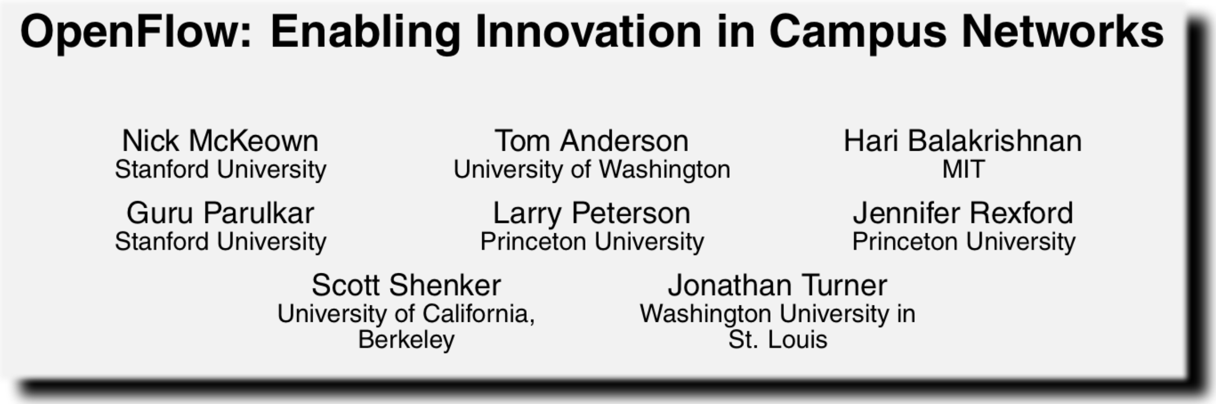


Ethane: Taking Control of the Enterprise

Martìn Casado, Michael J. Freedman,
Justin Pettit, Jianying Luo,
and Nick McKeown
Stanford University

Scott Shenker
U.C. Berkeley and ICSI

This paper presents Ethane, a new network architecture for the enterprise. Ethane allows managers to define a single network-wide fine-grain policy, and then enforces it directly. Ethane couples extremely simple flow-based Ethernet switches with a centralized controller that manages the admittance and routing of flows. While radical, this design is backwards-compatible with existing hosts and switches.



OpenFlow: Enabling Innovation in Campus Networks

Nick McKeown
Stanford University

Tom Anderson
University of Washington

Hari Balakrishnan
MIT

Guru Parulkar
Stanford University

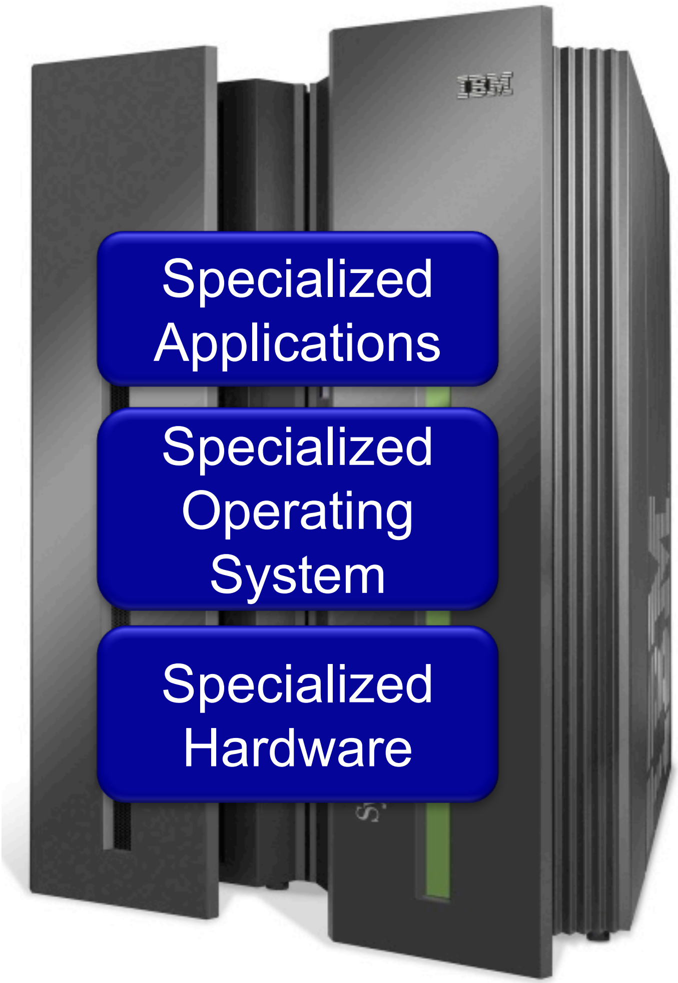
Larry Peterson
Princeton University

Jennifer Rexford
Princeton University

Scott Shenker
University of California,
Berkeley

Jonathan Turner
Washington University in
St. Louis

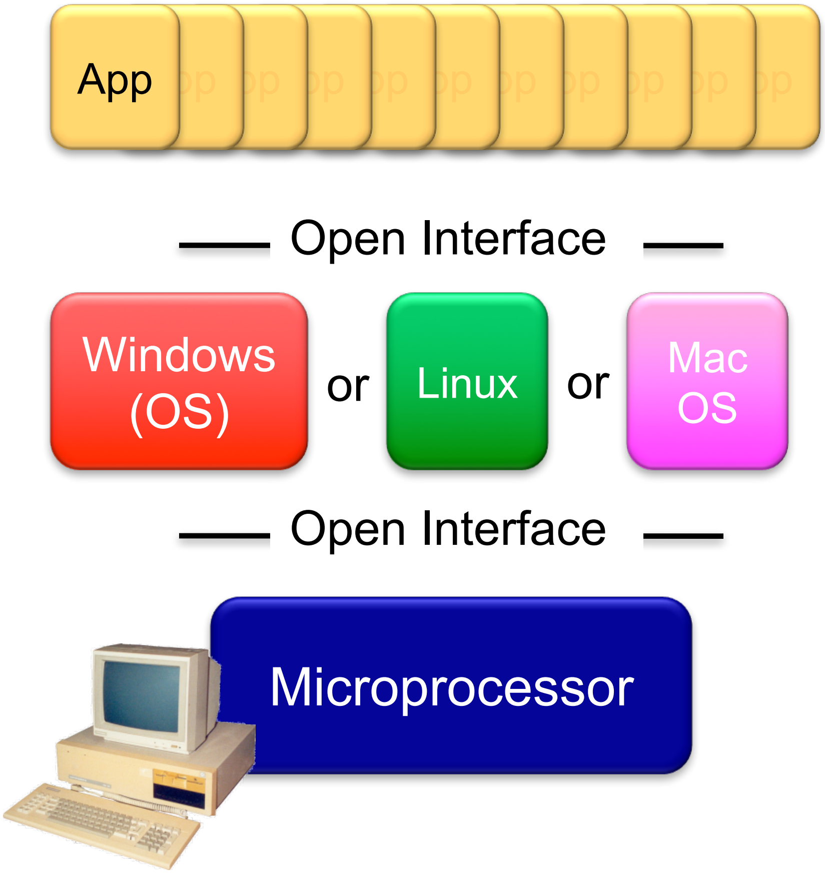
(slide by Nick McKeown, Stanford University)



Specialized Applications

Specialized Operating System

Specialized Hardware



App

Open Interface

Windows (OS) or Linux or Mac OS

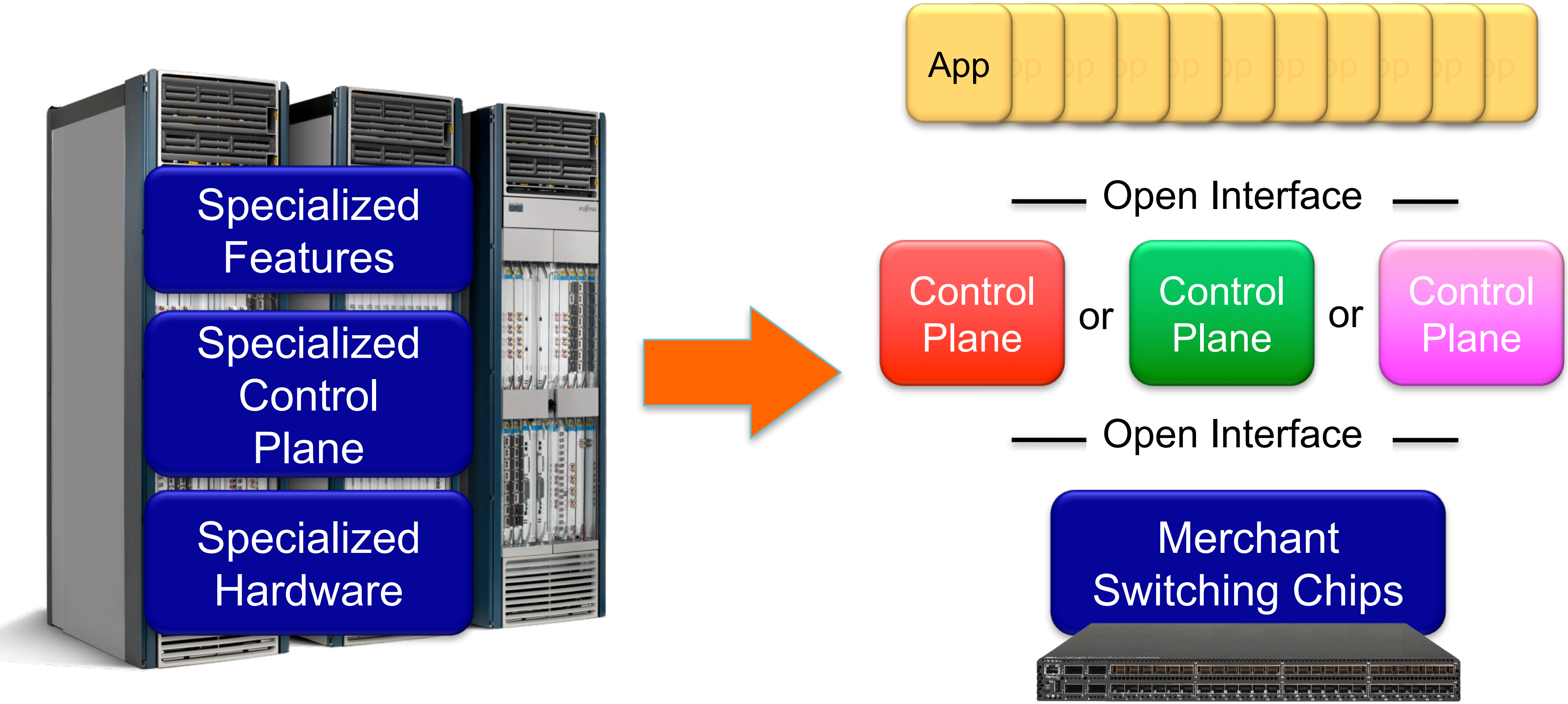
Open Interface

Microprocessor

Vertically integrated
Closed, proprietary
Slow innovation
Small industry

Horizontal
Open interfaces
Rapid innovation
Huge industry

(slide by Nick McKeown, Stanford University)



Specialized Features

Specialized Control Plane

Specialized Hardware

App

Open Interface

Control Plane or Control Plane or Control Plane

Open Interface

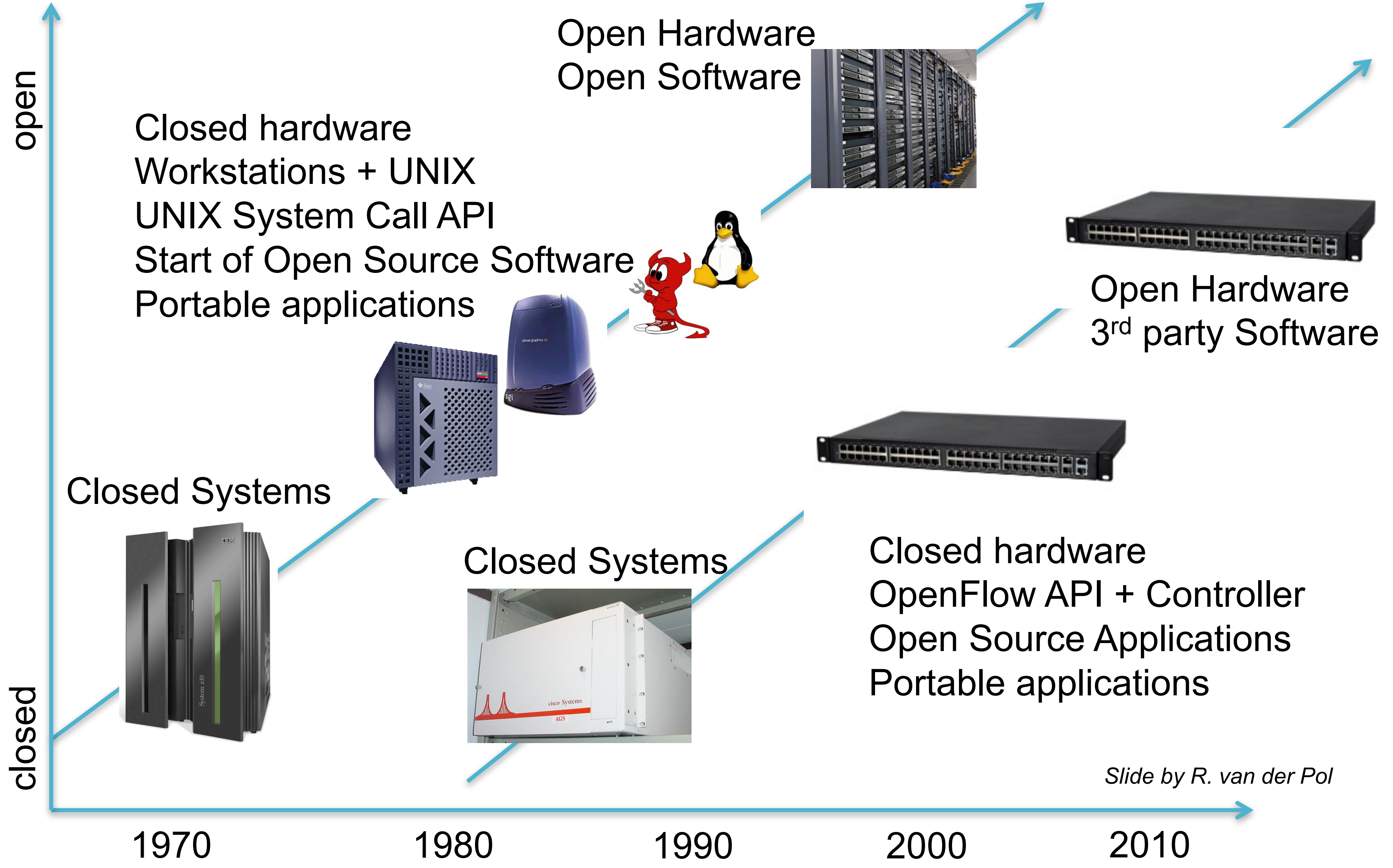
Merchant Switching Chips

Vertically integrated
Closed, proprietary
Slow innovation

→

Horizontal
Open interfaces
Rapid innovation

Computing vs Networking



Closed Systems

Closed hardware
Workstations + UNIX
UNIX System Call API
Start of Open Source Software
Portable applications

Open Hardware
Open Software

Closed Systems

Closed hardware
OpenFlow API + Controller
Open Source Applications
Portable applications

Open Hardware
3rd party Software

Slide by R. van der Pol

Disaggregation

From

Closed vendor proprietary routers/switches

- Vendor decides which features to support
- Vendor decides when these feature become available
- Dependent on innovation strength of vendor

To

Split between hardware and firmware (disaggregation)

- Choose best vendor for hardware and best vendor for firmware
- Use open source firmware (more control over features and innovation)

“Protocol Soup”

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

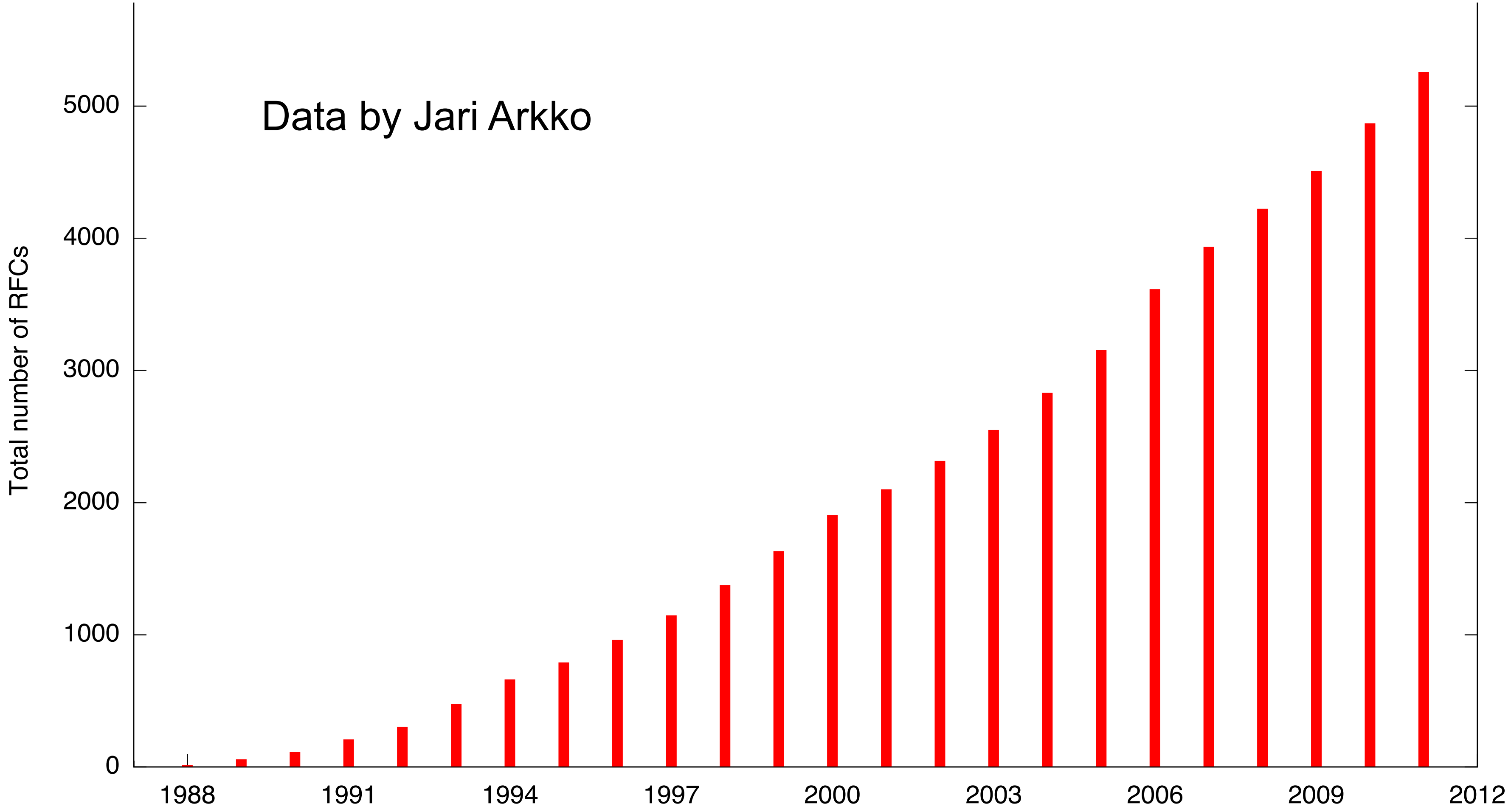
Exponential growth in network protocol standards.

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?

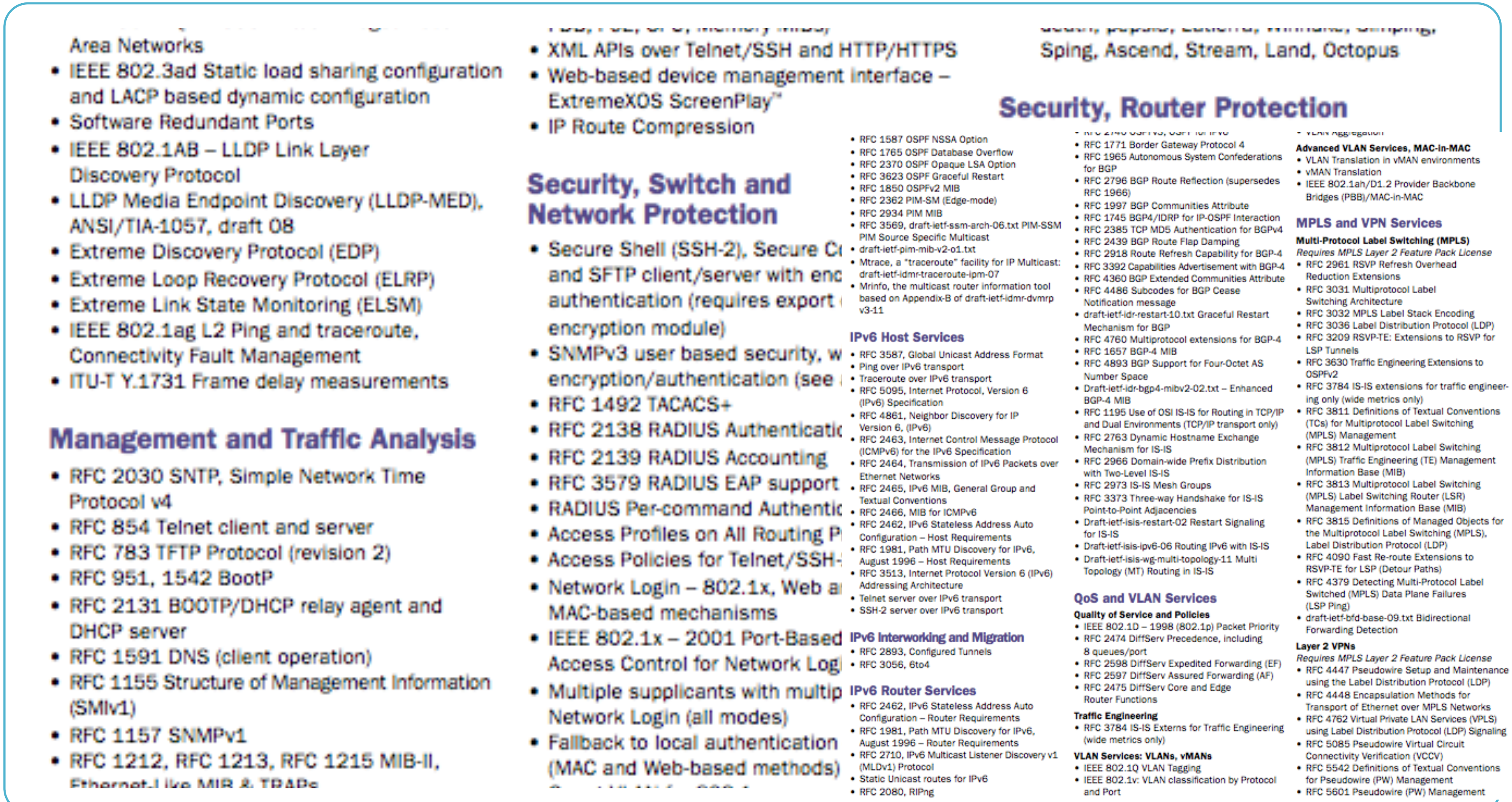
Not really, original version amended by at least 14 additional standards.

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, Ethernet-Like MIB & TRAPs

- 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 Authenticatio
- 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 2740 OSPFv3, OSPF for IPv6
- 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
- VLAN Aggregation

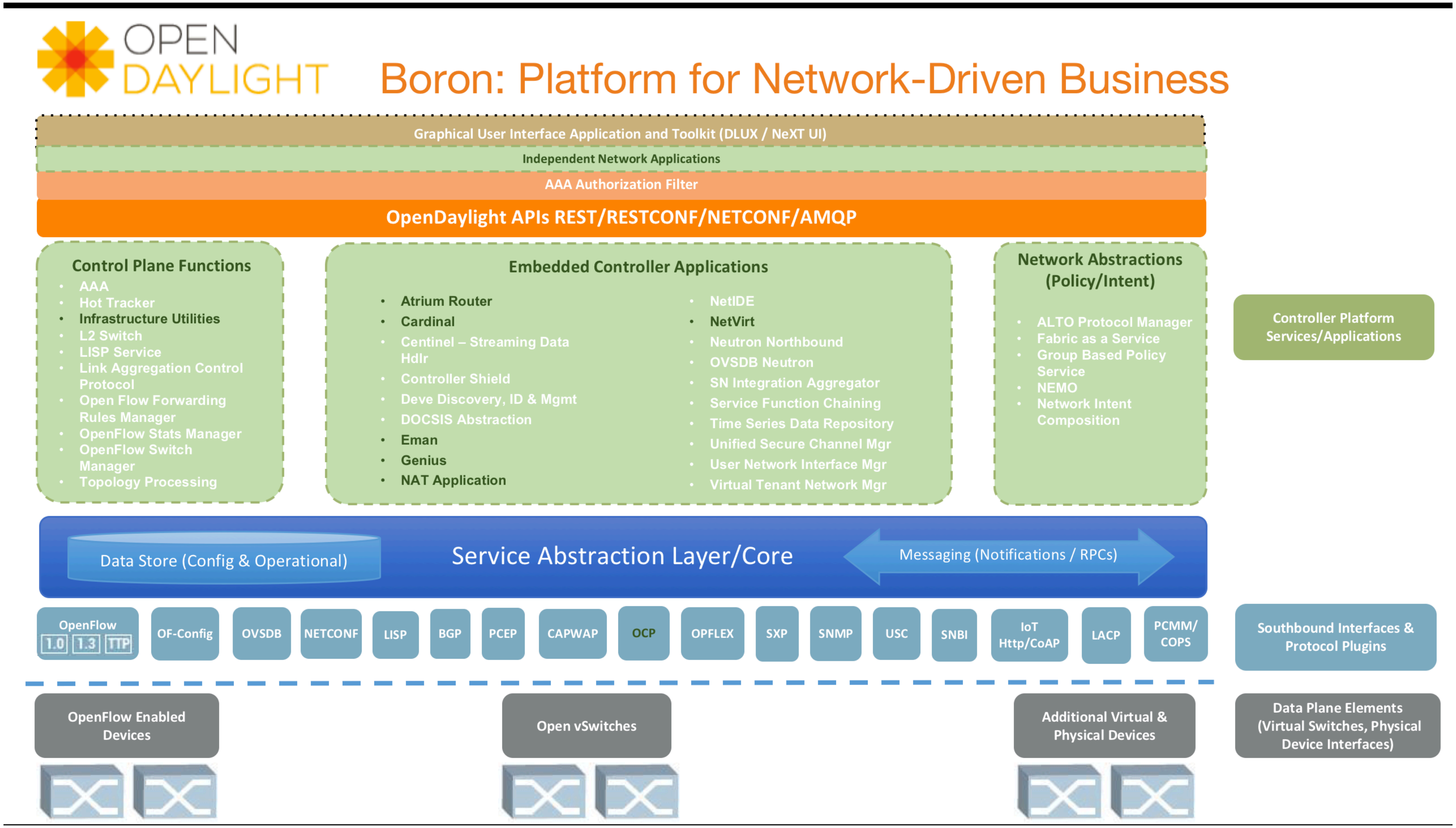
Advanced VLAN Services, MAC-in-MAC
- VLAN Translation in vMAN environments
- vMAN Translation
- IEEE 802.1ah/D1.2 Provider Backbone Bridges (PBB)/MAC-in-MAC

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

OpenDaylight Architecture



OPEN DAYLIGHT

Boron: Platform for Network-Driven Business

Graphical User Interface Application and Toolkit (DLUX / NeXT UI)

Independent Network Applications

AAA Authorization Filter

OpenDaylight APIs REST/RESTCONF/NETCONF/AMQP

Control Plane Functions

- AAA
- Hot Tracker
- Infrastructure Utilities
- L2 Switch
- LISP Service
- Link Aggregation Control Protocol
- Open Flow Forwarding Rules Manager
- OpenFlow Stats Manager
- OpenFlow Switch Manager
- Topology Processing

Embedded Controller Applications

- Atrium Router
- Cardinal
- Centinel – Streaming Data Hdlr
- Controller Shield
- Deve Discovery, ID & Mgmt
- DOCSIS Abstraction
- Eman
- Genius
- NAT Application
- NetIDE
- NetVirt
- Neutron Northbound
- OVSDB Neutron
- SN Integration Aggregator
- Service Function Chaining
- Time Series Data Repository
- Unified Secure Channel Mgr
- User Network Interface Mgr
- Virtual Tenant Network Mgr

Network Abstractions (Policy/Intent)

- ALTO Protocol Manager
- Fabric as a Service
- Group Based Policy Service
- NEMO
- Network Intent Composition

Controller Platform Services/Applications

Data Store (Config & Operational)

Service Abstraction Layer/Core

Messaging (Notifications / RPCs)

OpenFlow 1.0 1.3 TTP

OF-Config

OVSDB

NETCONF

LISP

BGP

PCEP

CAPWAP

OCP

OPFLEX

SXP

SNMP

USC

SNBI

IoT Http/CoAP

LACP

PCMM/ COPS

Southbound Interfaces & Protocol Plugins

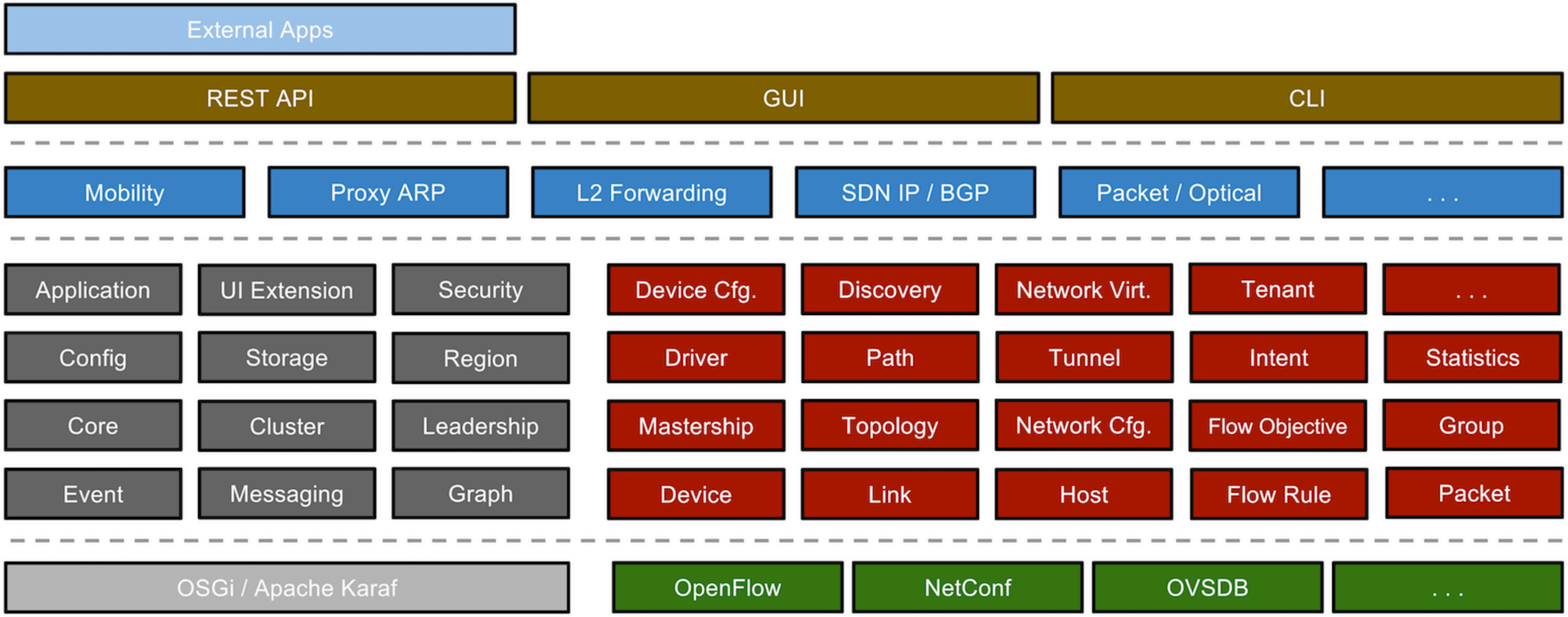
OpenFlow Enabled Devices

Open vSwitches

Additional Virtual & Physical Devices

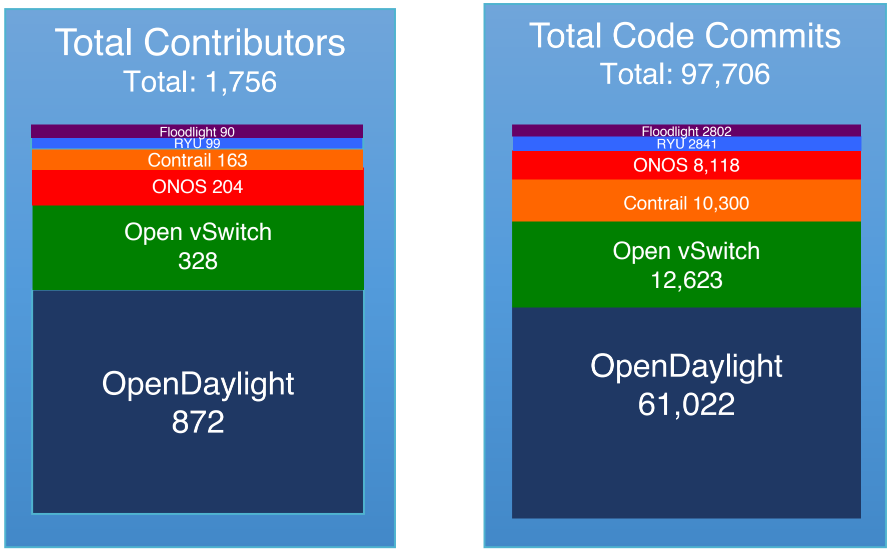
Data Plane Elements (Virtual Switches, Physical Device Interfaces)

ONOS Architecture

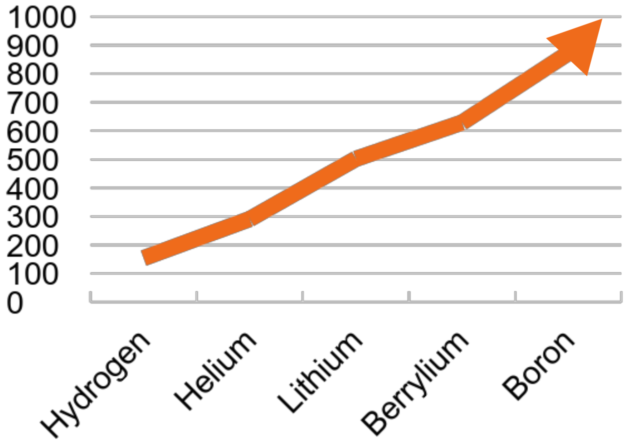


Largest Open Networking Dev Community

Open Networking / SDN Cumulative Contributions



OpenDaylight Contributors by Release



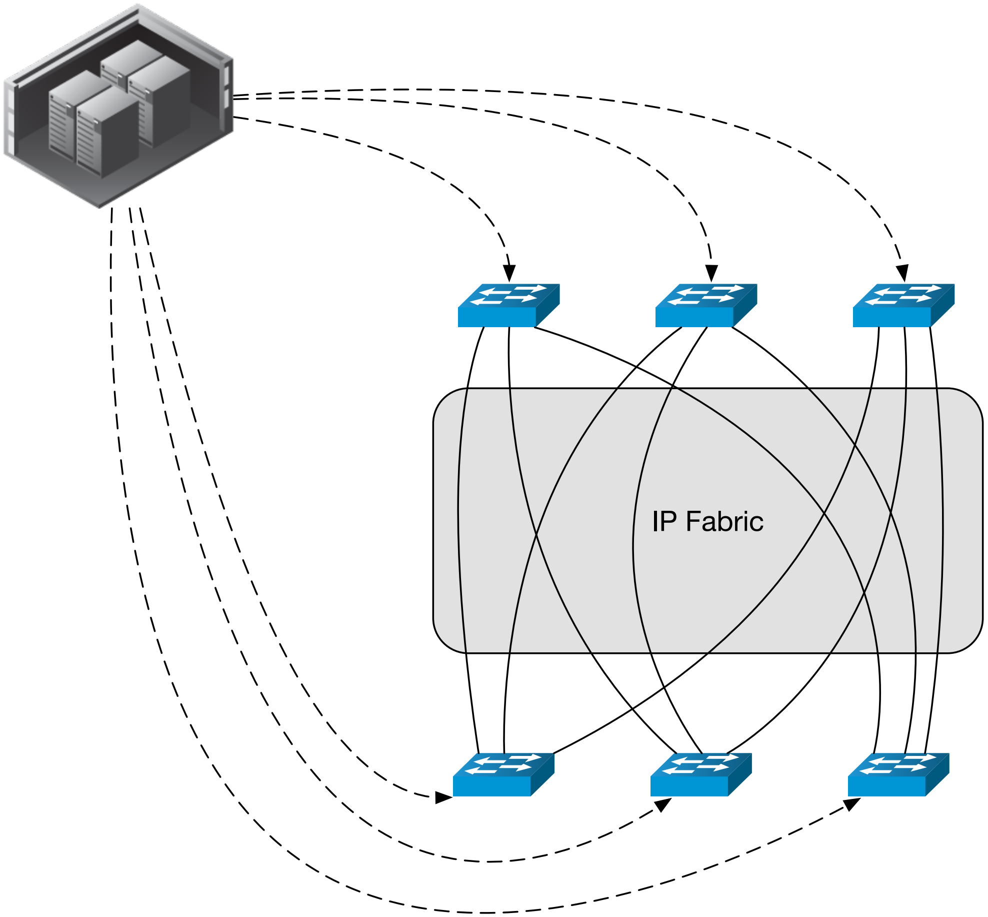
SDN Approaches

Underlay/Overlay networking

Logically centralised control of distributed control plane protocols

Logically centralised control of programmable dataplanes

Underlay/Overlay Networking



Underlay/Overlay Networking

Used in data centres, but also in WAN

VXLAN often used as tunneling protocol

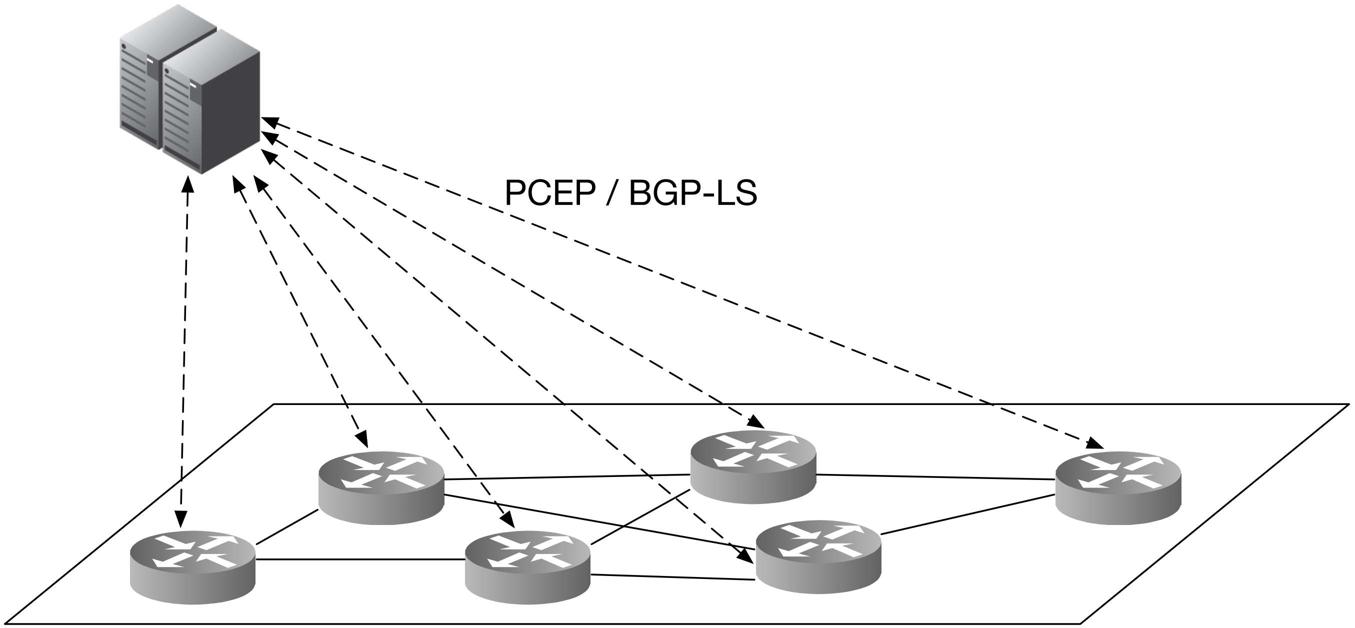
Open vSwitch and hypervisors at the edges of the tunnels

Virtualisation and multi-tenancy support

Controller often uses vendor proprietary protocols and APIs

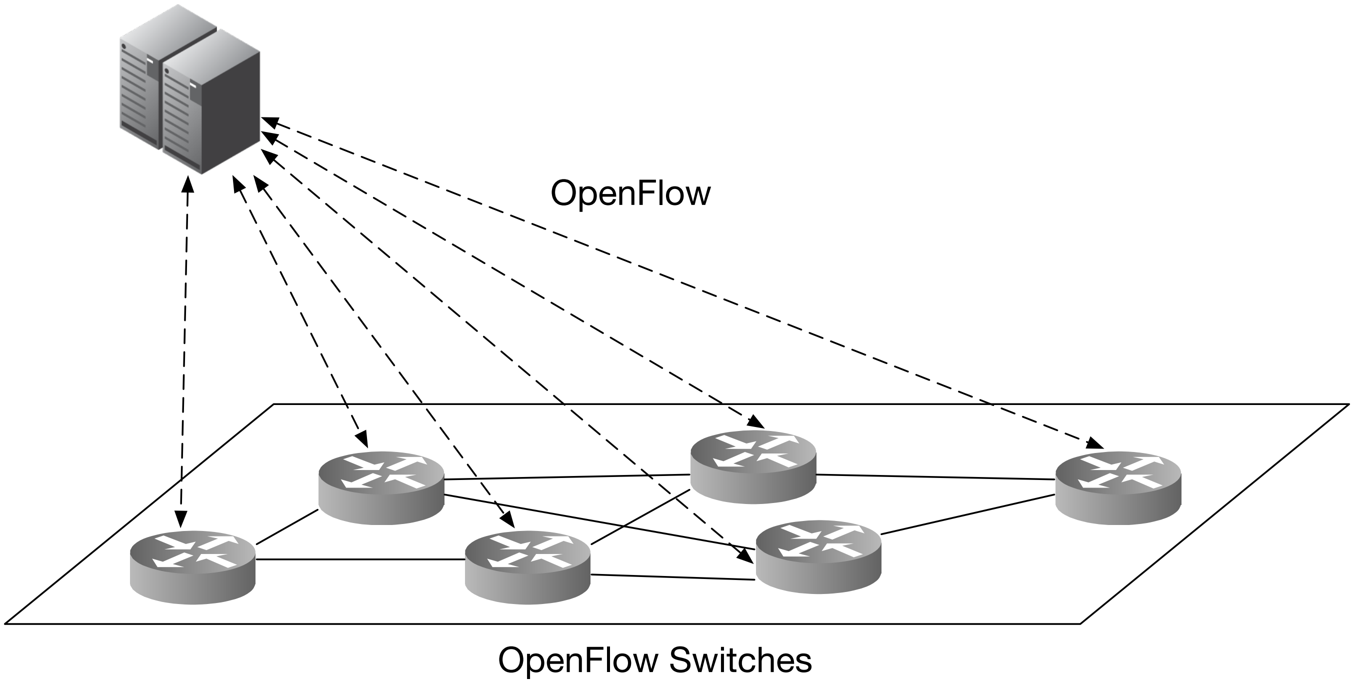
Many vendors offer solutions in this space

Combination of Centralised Control and Distributed Routing/Switching Protocols



MPLS / Segment Routing

OpenFlow Model



The hyperscales and SDN

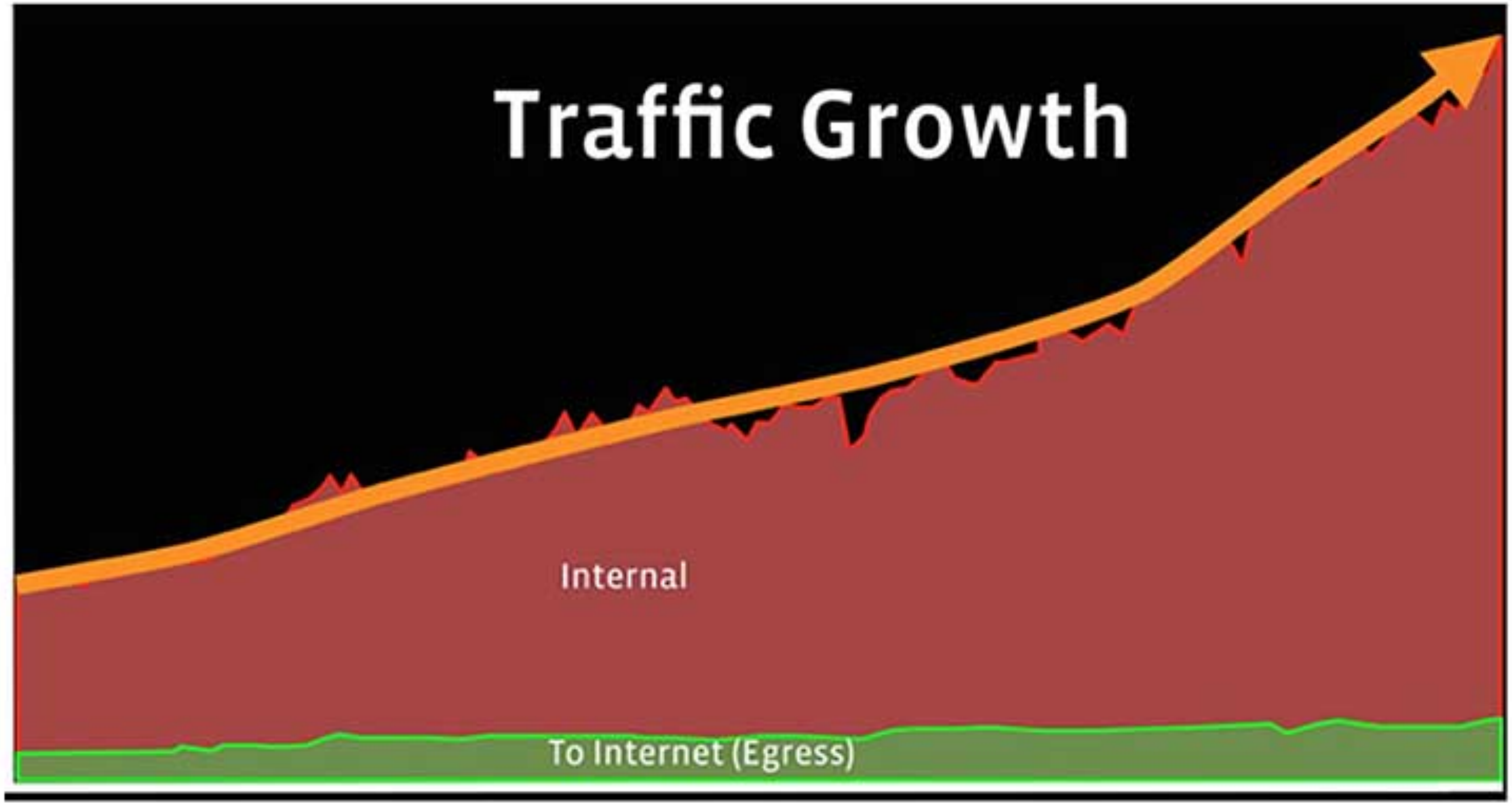
- **Common Elements:**
  - Estimated total amount of servers: ~ 1 million
  - Servers per data centre (estimated): 50,000 – 100,000
  - Combination of central control (Traffic Engineering) and distributed routing protocols
  - Disaggregated custom built switches
  - Homebuilt software, some of it open sourced

Facebook Old Design

Problems with previous (cluster + TOR) design

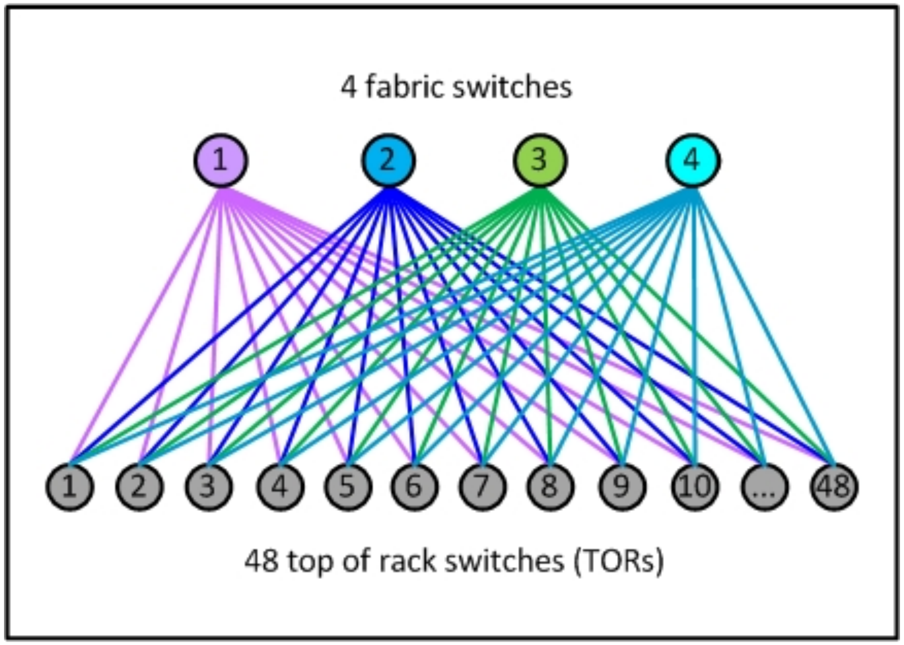
- **Many servers in a cluster connected to TOR switch**
- **Needed high-end switches**
- **Only a few vendors could deliver them**
- **Required extensive platform-specific hardware and software knowledge to operate and troubleshoot**
- **Failures could have significant impact**
- **Limited bandwidth between clusters**

Facebook Traffic Growth



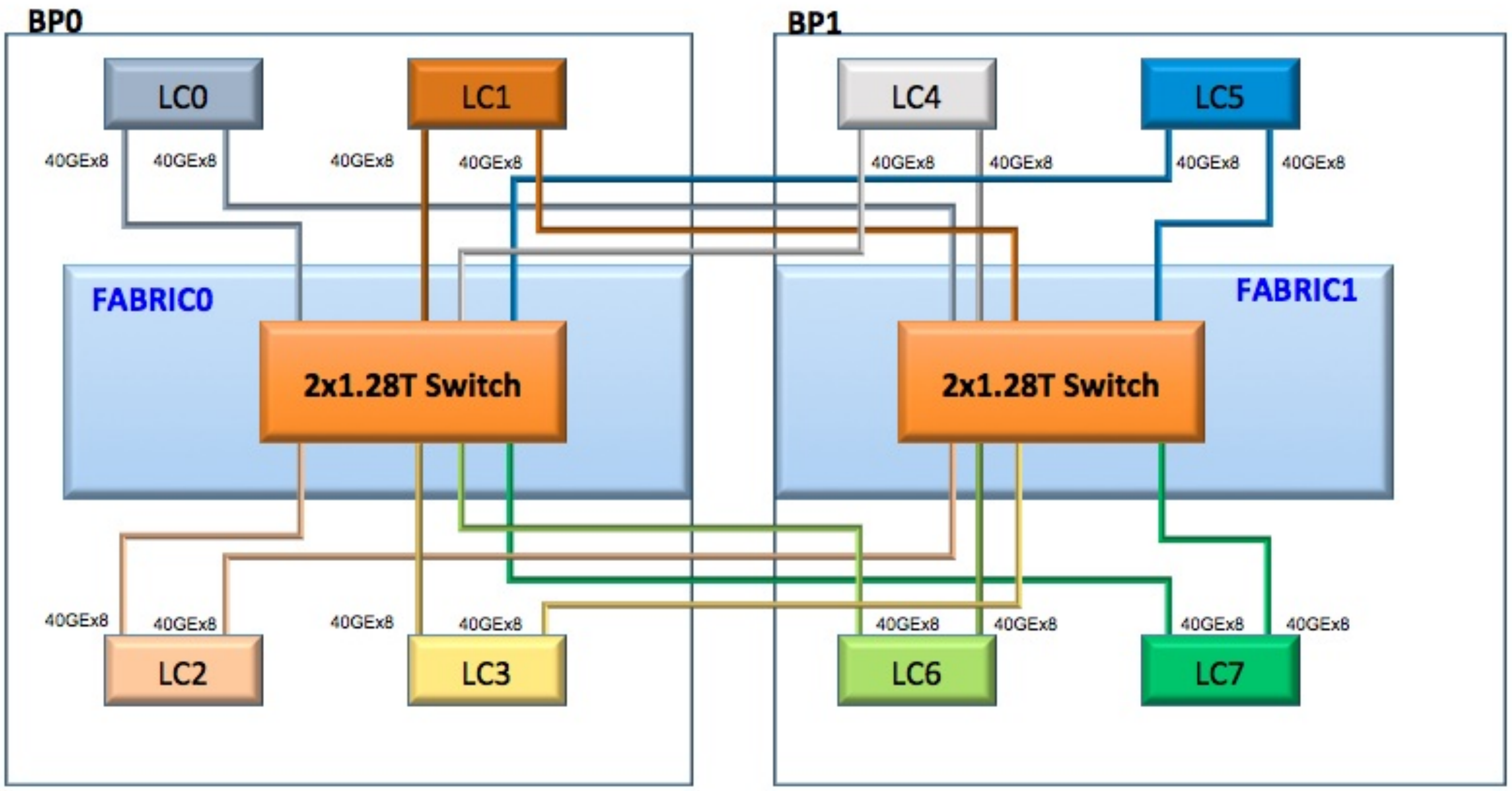
Facebook New Design (fabric and pods)

- Small identical units: pods
- TOR: 4x 40G, 10G to servers
- (guess) 16x48=768 servers/pod
- Fabric switch: 6-pack

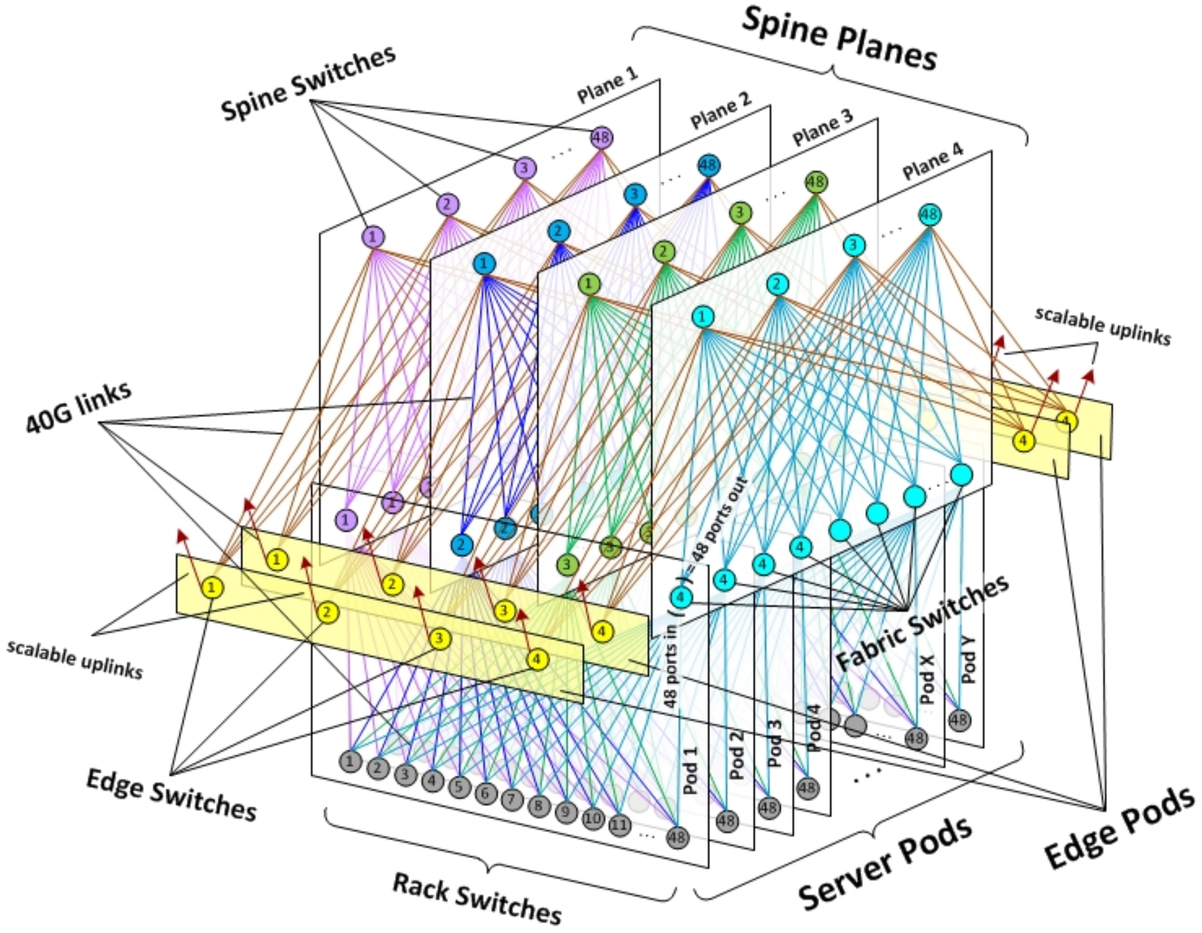


Facebook 6-Pack: 128 (8 x 16) x 40G Ports

6-pack Architecture



Facebook Datacenter Architecture



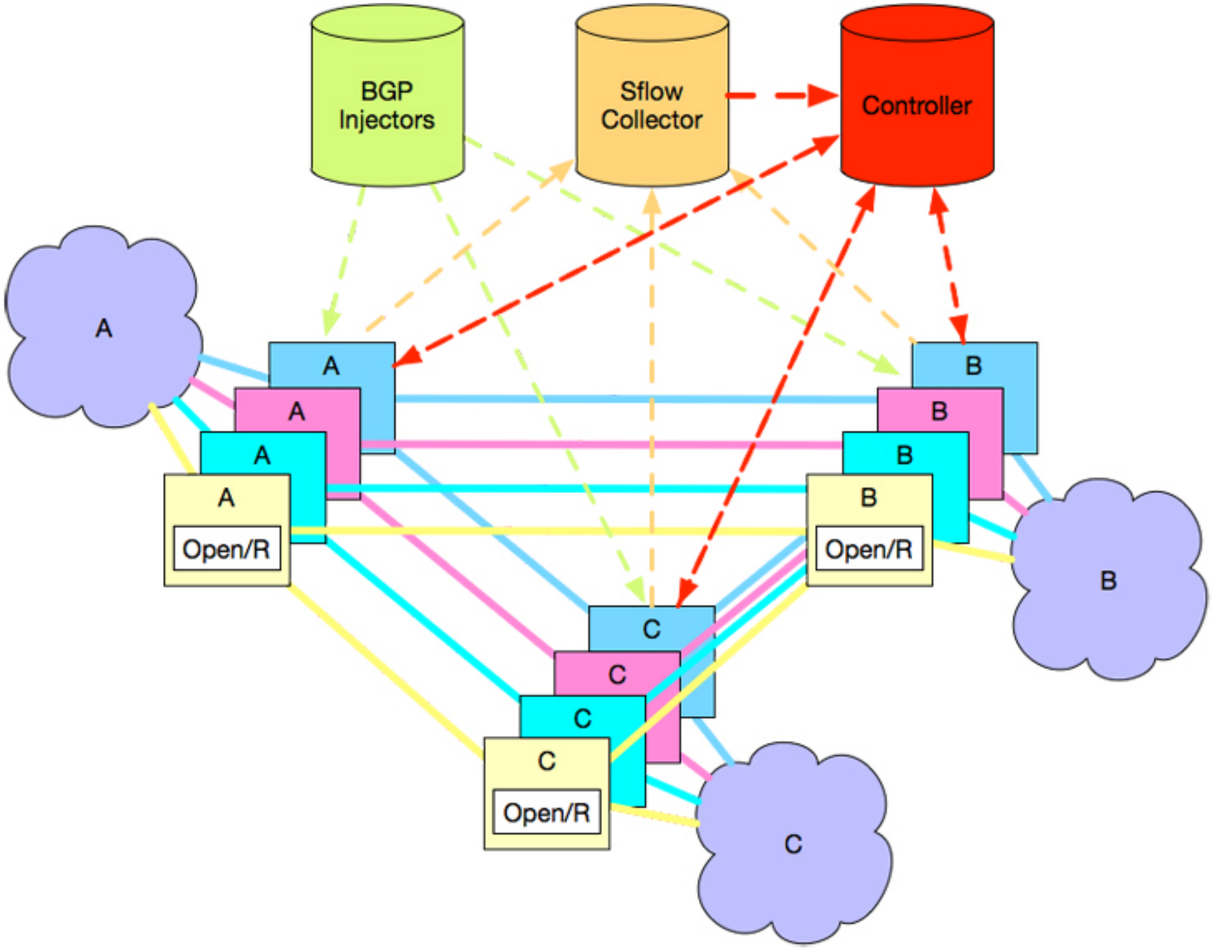
Facebook Fabric

- **Dual-stack L3 from TOR to edge**
- **BGP4 using minimal amount of protocol features**
- **ECMP**
- **Routing design minimises use of RIB and FIB resources**
- **Split control (server on the switch connected to central controller)**
- **“Distributed control, centralized override”**

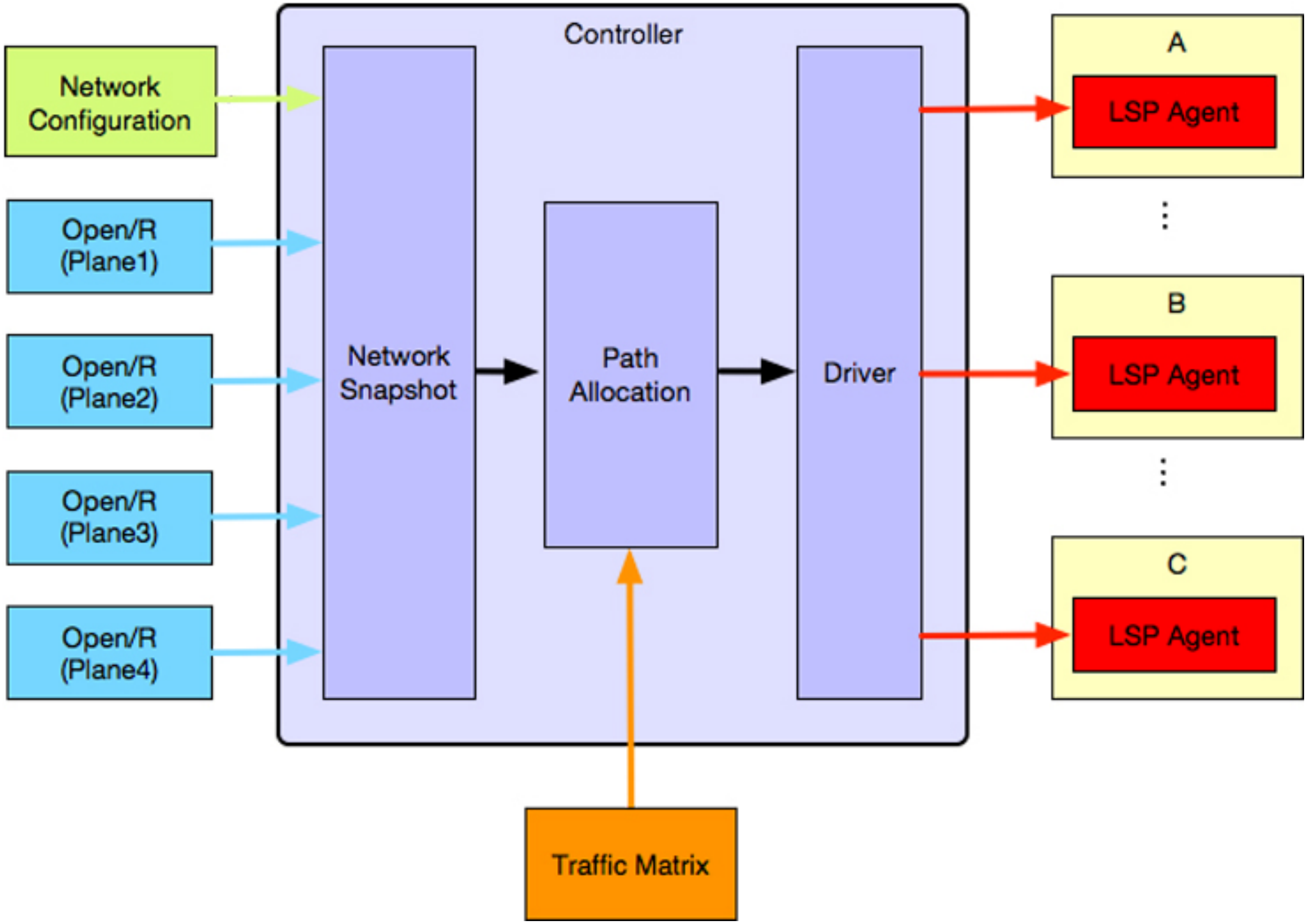
Facebook Express Backbone (EBB)

- **MPLS Segment Routing**
- **Facebook Open/R as IGP, integrated with central controller**
- **Physical network topology split into four ”planes”**
- **Centralised ensemble of BGP-based route injectors**
- **Traffic Engineering (TE) controller computes optimal paths**
- **sFlow collector feeds active demands into TE controller**
- **LSP agents on the network devices interface with forwarding tables on behalf of the TE controller**

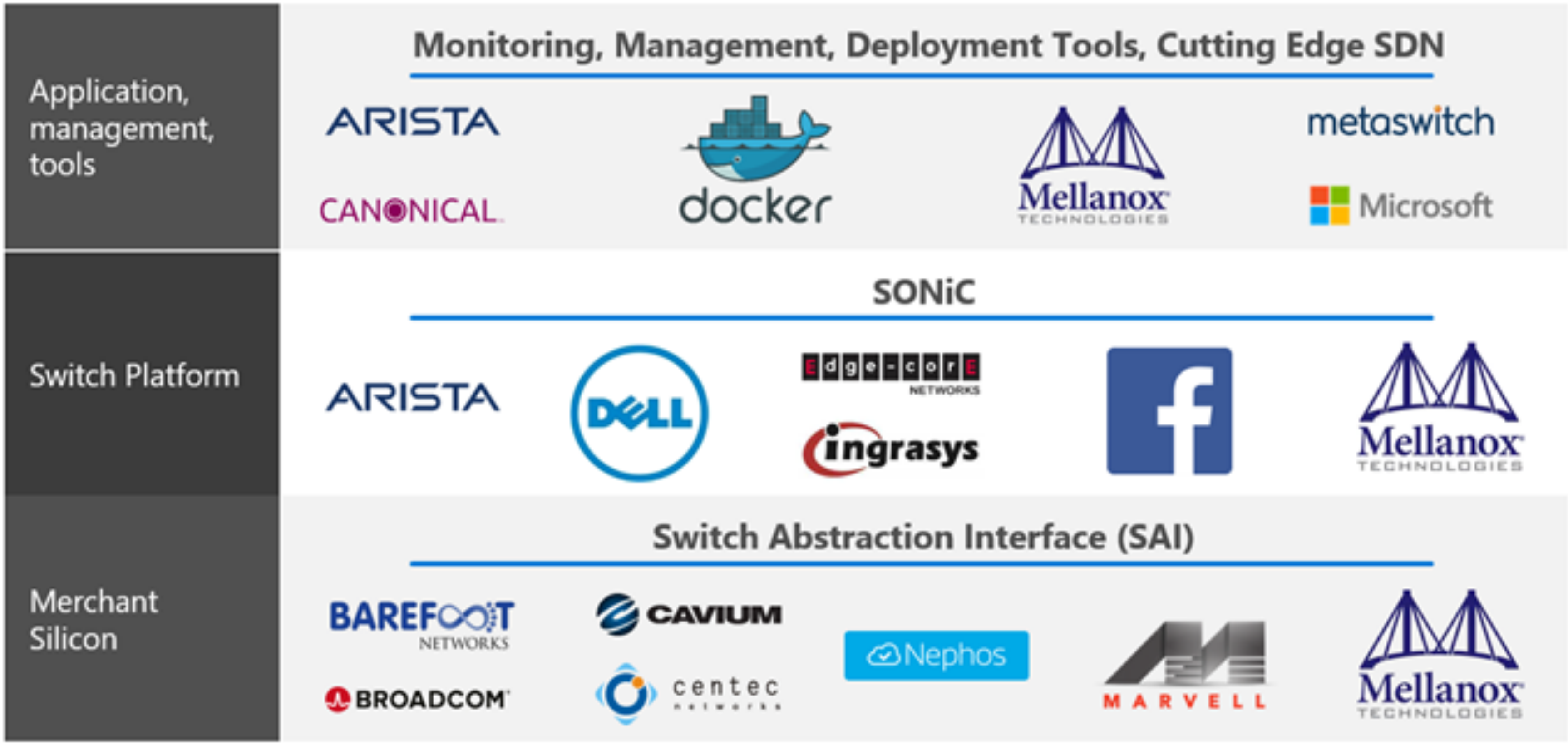
Facebook EBB Architecture



Facebook EBB TE Controller



Microsoft SAI & SONiC



Application, management, tools

Monitoring, Management, Deployment Tools, Cutting Edge SDN

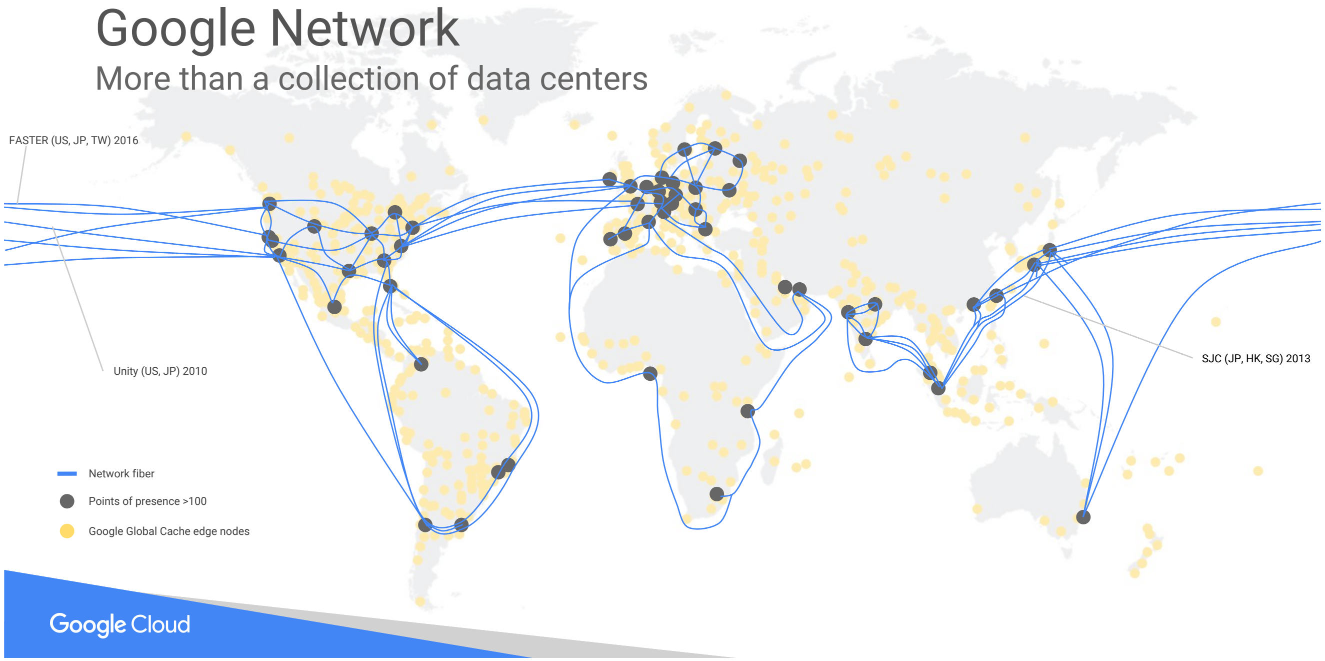
Switch Platform

SONiC

Merchant Silicon

Switch Abstraction Interface (SAI)

Google Network

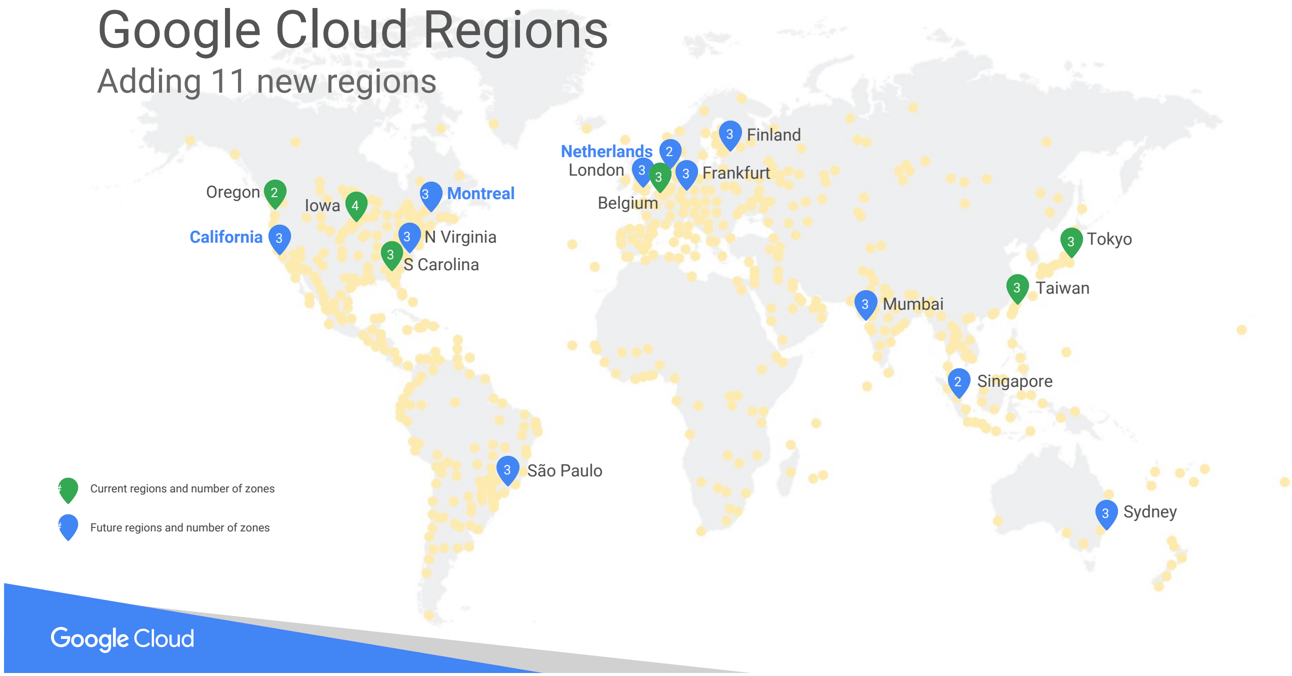


Google Network

More than a collection of data centers

Following slides are by Amin Vahdat, keynote at ONS 2017

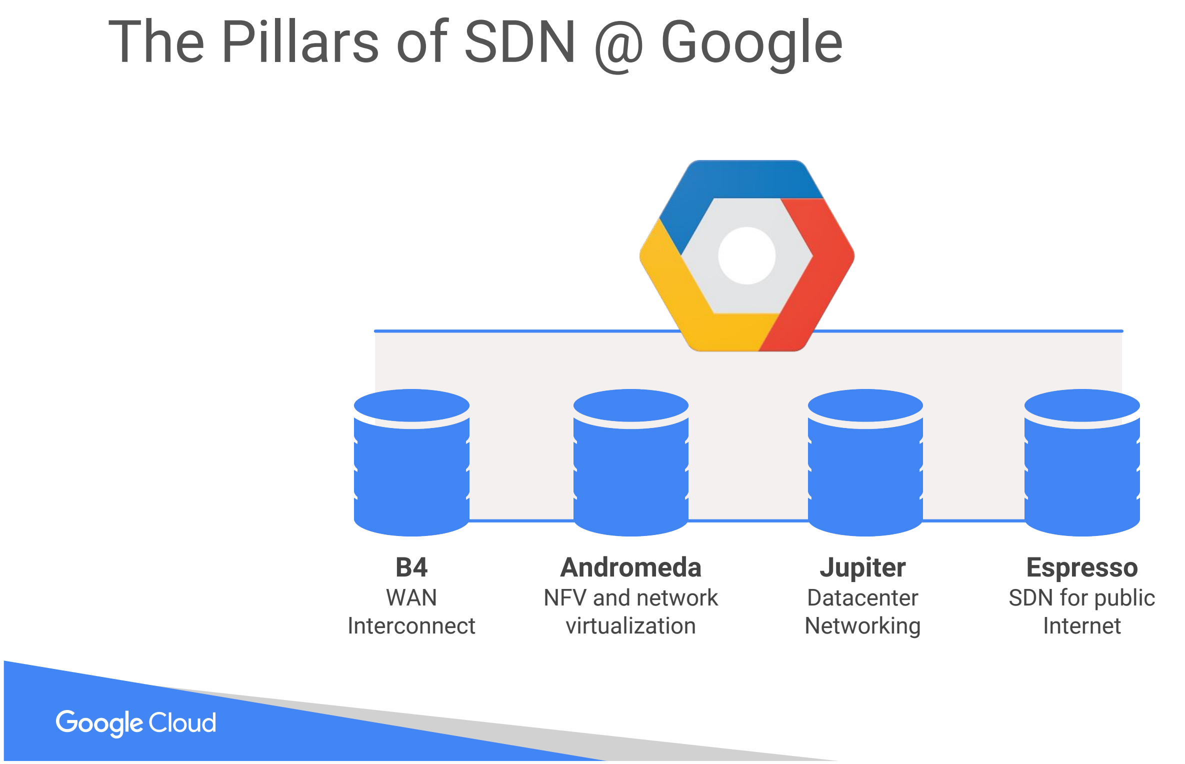
Google Cloud Regions



Google Cloud Regions

Adding 11 new regions

Google Pillars of SDN



The Pillars of SDN @ Google

B4
WAN Interconnect

Andromeda
NFV and network virtualization

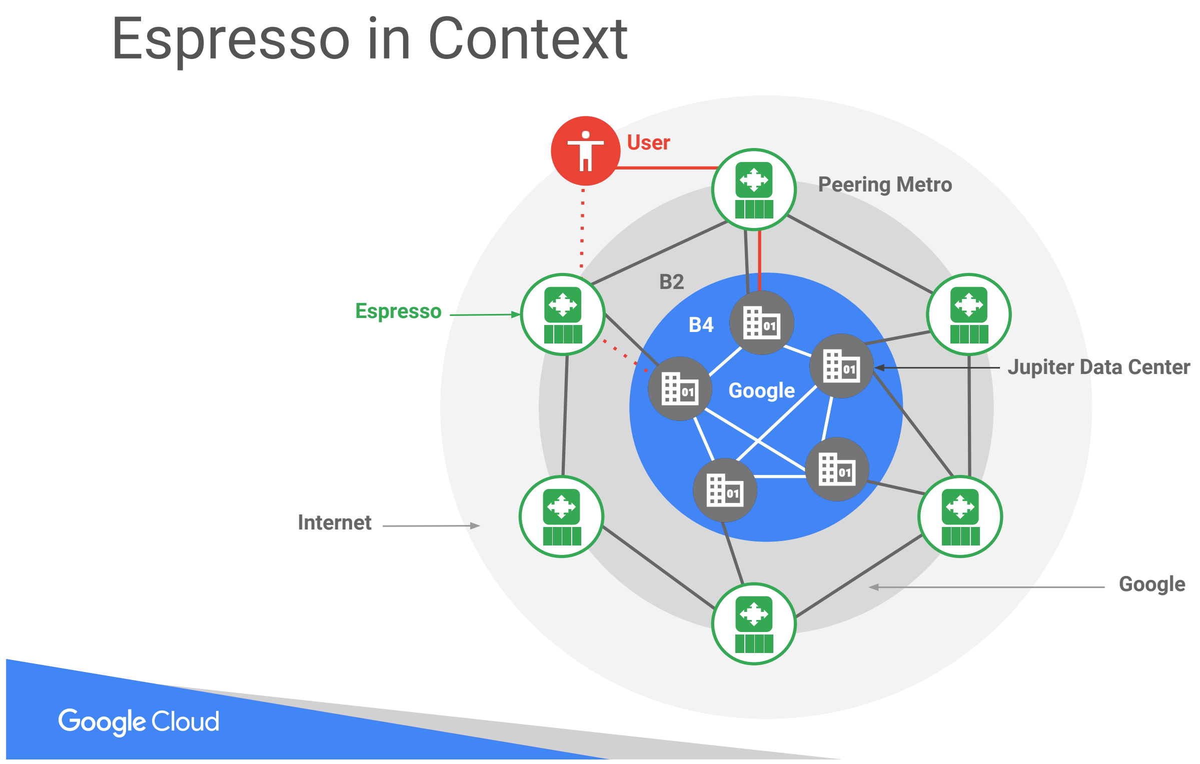
Jupiter
Datacenter Networking

Espresso
SDN for public Internet

Google SDN

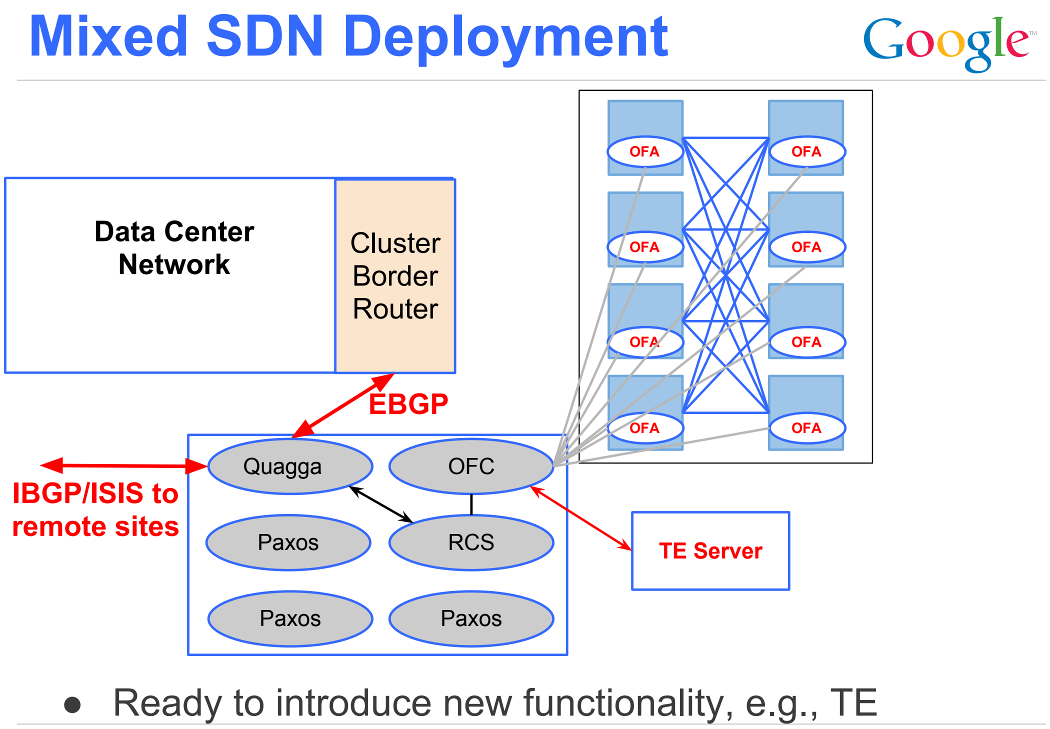
- Internal traffic (machine to machine) grows faster than user traffic (similar to Facebook internal/external growth)
- Google carries 25% of internet traffic

Google Networking



Espresso in Context

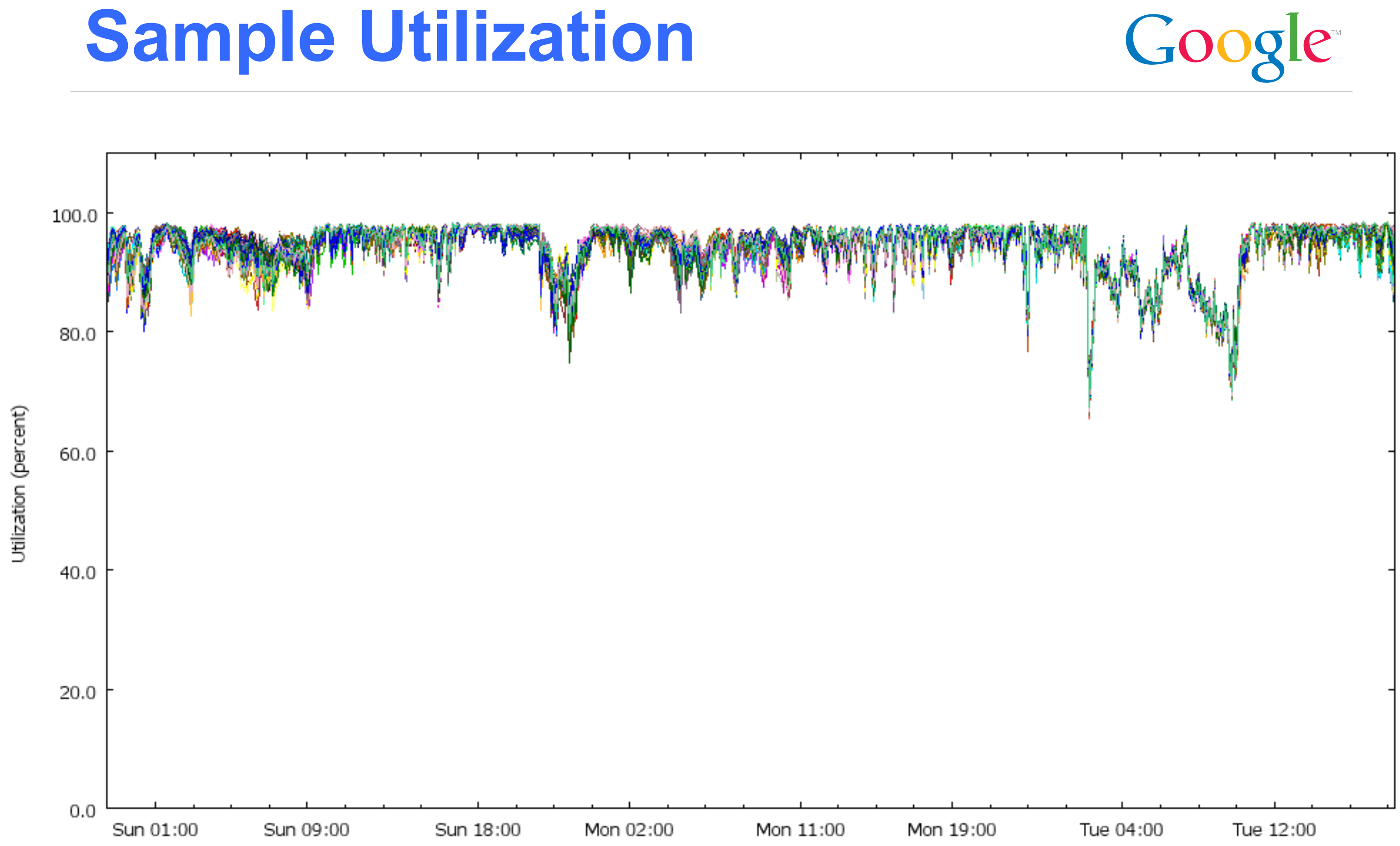
Google B4 Network



Mixed SDN Deployment

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

Almost 100% Link Utilization

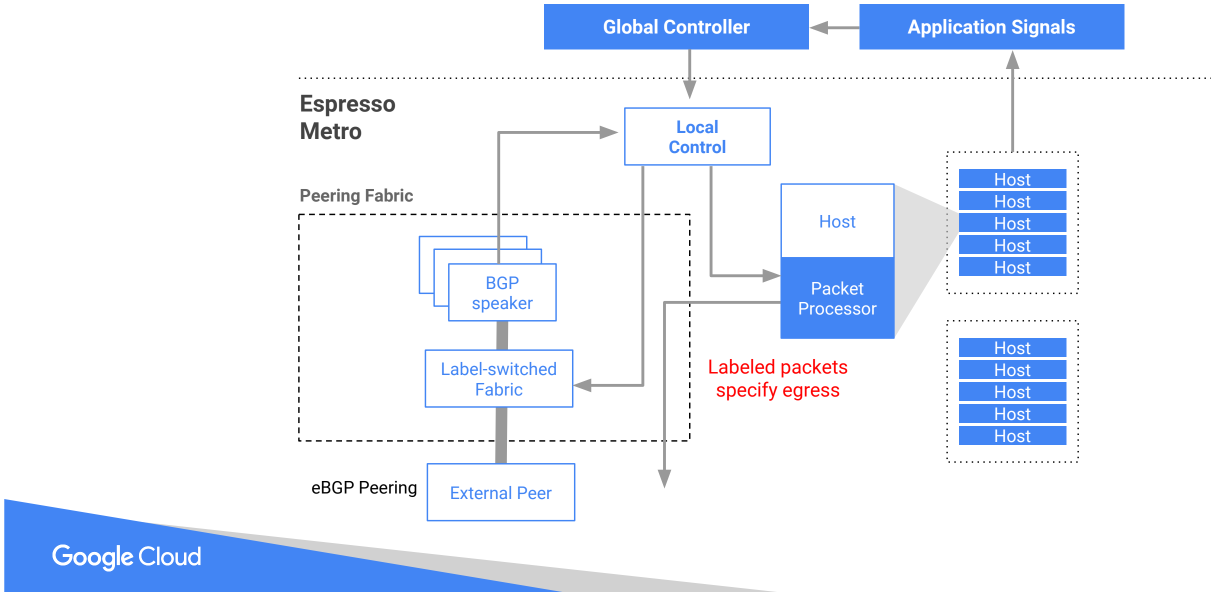


Sample Utilization

Google

Google Espresso Atchitecture

Espresso Architecture Overview



Network Research Topics at SURFnet

- **Networking for containers and serverless applications**
- **SMART NICs (policy enforcement and performance - P4 & eBPF)**
- **Programmable Switches (P4)**
- **SnapRoute (L2/L3) & Open Source Network Operating Systems**

P4 Language

P4: Programming Protocol-Independent Packet Processors

Domain Specific Language for network protocols

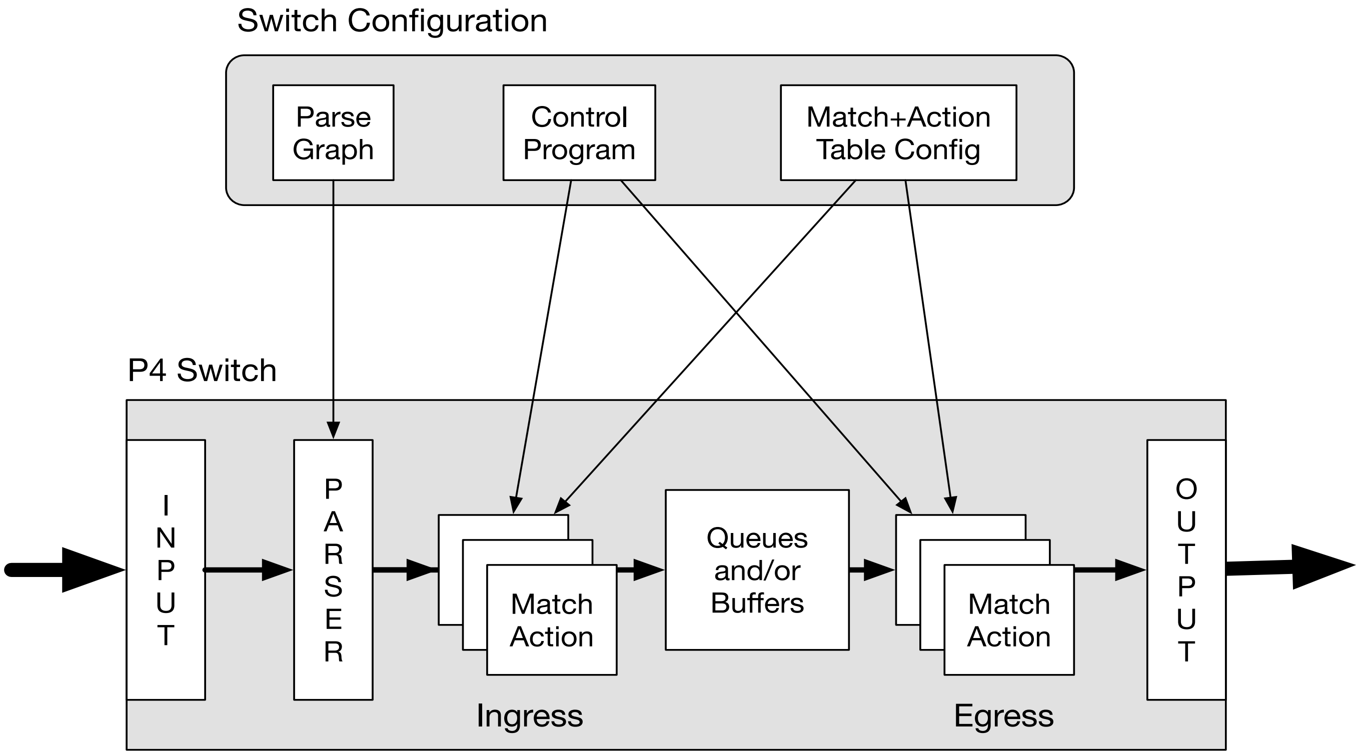
Takes the OpenFlow match/action concepts much further

P4 program defines headers, parser and lookup tables

P4 program → P4 compiler → target code → P4 switch/NIC

Target code is loaded on P4 hardware/software switch/NIC

P4 Switch



Source: The P4 Language Specification Version 1.0.2

P4 Elements

P4 consists of three elements:

- **header definitions (packet headers and metadata)**
- **lookup tables & actions (match/action)**
- **packet parser state machine & checksum field lists**

P4 Header Definition Examples

```
header_type ethernet_t {
    fields {
        dstAddr : 48;
        srcAddr : 48;
        etherType : 16;
    }
}

header ethernet_t ethernet;
```

```
header_type ipv4_t {
    fields {
        version : 4;
        ihl : 4;
        diffserv : 8;
        totalLen : 16;
        identification : 16;
        flags : 3;
        fragOffset : 13;
        ttl : 8;
        protocol : 8;
        hdrChecksum : 16;
        srcAddr : 32;
        dstAddr: 32;
    }
}
```

P4 Metadata Example

```
header_type ingress_metadata_t {
    fields {
         ingress_port: 9;
         packet_length: 16;
         ingress_global_tstamp: 48;
        egress_spec: 16;
        queue_id: 9;
    }
}

header ingress_metadata ingress_metadata;
```

P4 Parser

```
parser start {
    return parse_ethernet;
}

parser parse_ethernet {
    extract(ethernet);
    return select(latest.etherType) {
        ETHERTYPE_IPV4 : parse_ipv4;
        default: ingress;
    }
}

parser parse_ipv4 {
    return select(lastest.<some IPv4 field>) {
       <some field value>: do_something;
       default ingress;
}
```

P4 L2 Switch Table Example

```
table dmac {
    reads {
        ethernet.dstAddr: exact;
    }
    actions {
        forward;
        broadcast;
    }
    size : 512;
}
```

P4 IPv4 FIB Table Actions

```
action forward(port) {
        modify_field(standard_metadata.egress_port, port);
}
```

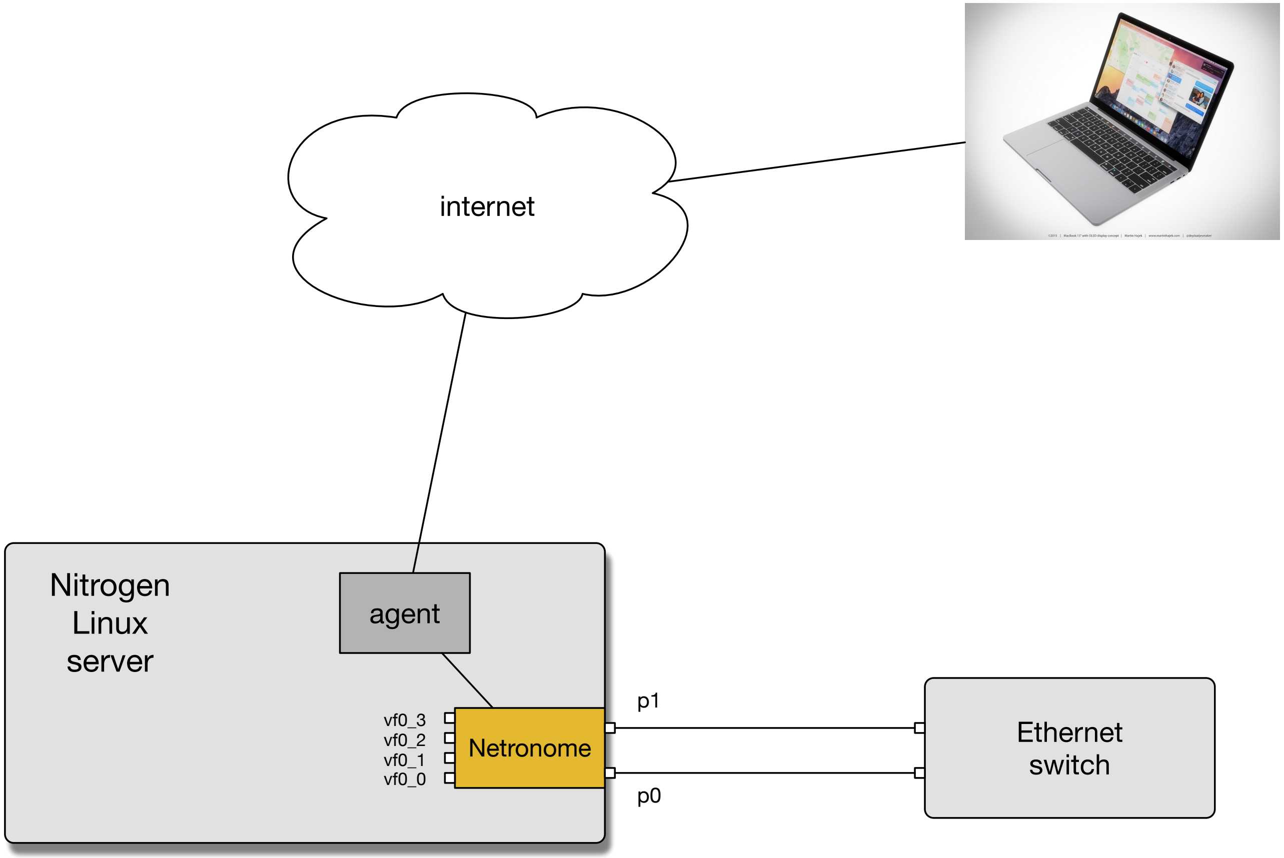
What Happened To OpenFlow?

- Created a whole new market of white label switches
- Many vendors offer “SDN” solutions
- All hyperscales have networks based on the original OpenFlow concepts

Amin Vahdat
(Fellow & Technical Lead for Networking, Google)
Keynote speech at ONS 2017:

“To me the question of whether software defined networking is a good idea or not is closed. Software defined networking is how we do networking.”

Demo Setup


internet
Nitrogen Linux server
agent
vf0_3
vf0_2
vf0_1
vf0_0
Netronome
p1
p0
Ethernet switch

Workshop

Dataplane Accelaration Developer Day
DXDD Europe 2017
Workshop with hands-on Labs
Wed 7 June 2017
SURFnet, Utrecht

http://www.open-nfp.org/dxdd-europe-2017

Ronal van der Pol
Ronald.vanderPol@SURFnet.nl

WHAT SURF CAN DO