SDN/OpenFlow Overview

UWDC

Utrecht, 6 Maart 2013

Ronald van der Pol

Ronald.vanderPol@SURFnet.nl

Outline

- Why was OpenFlow developed?
- How does OpenFlow work?
- Standardisation (ONF)
- Some examples of how OpenFlow is used
  - Data Centre
  - Campus Networks
  - Google Data Network
- Some of the OpenFlow Players
- Conclusions

Why OpenFlow?

- OpenFlow is a form of Software Defined Networking (SDN)
- Enable network innovation (again)
- Reducing operational costs (OPEX)
- Alternative for “protocol soup”
- Applying computing model to networking

Enable Network Innovation

- OpenFlow was developed at Stanford University as part of Clean Slate program
- University network needs to have 24x7 availability
- Potential disruptive network tests impossible
- OpenFlow enables slicing the network in production and experimental part

OPEX in Networking

- Adding routers and switches to your network increases the operational cost
- Each new device needs to be configured manually via the CLI and its neighbours need to be configured too
- Firmware updates on routers and switches with slow CPUs takes a long time, but much faster on a centralised SDN computer cluster
- Changes usually involves configuration actions on all devices

OPEX in Computing

- Scales much better
- Adding servers to a computer grid or cloud cluster does not increase the operational cost
- Middleware software with centralized policy (OpenStack, etc) controls the servers
- Configure the software once and push the button to apply the changes to all servers

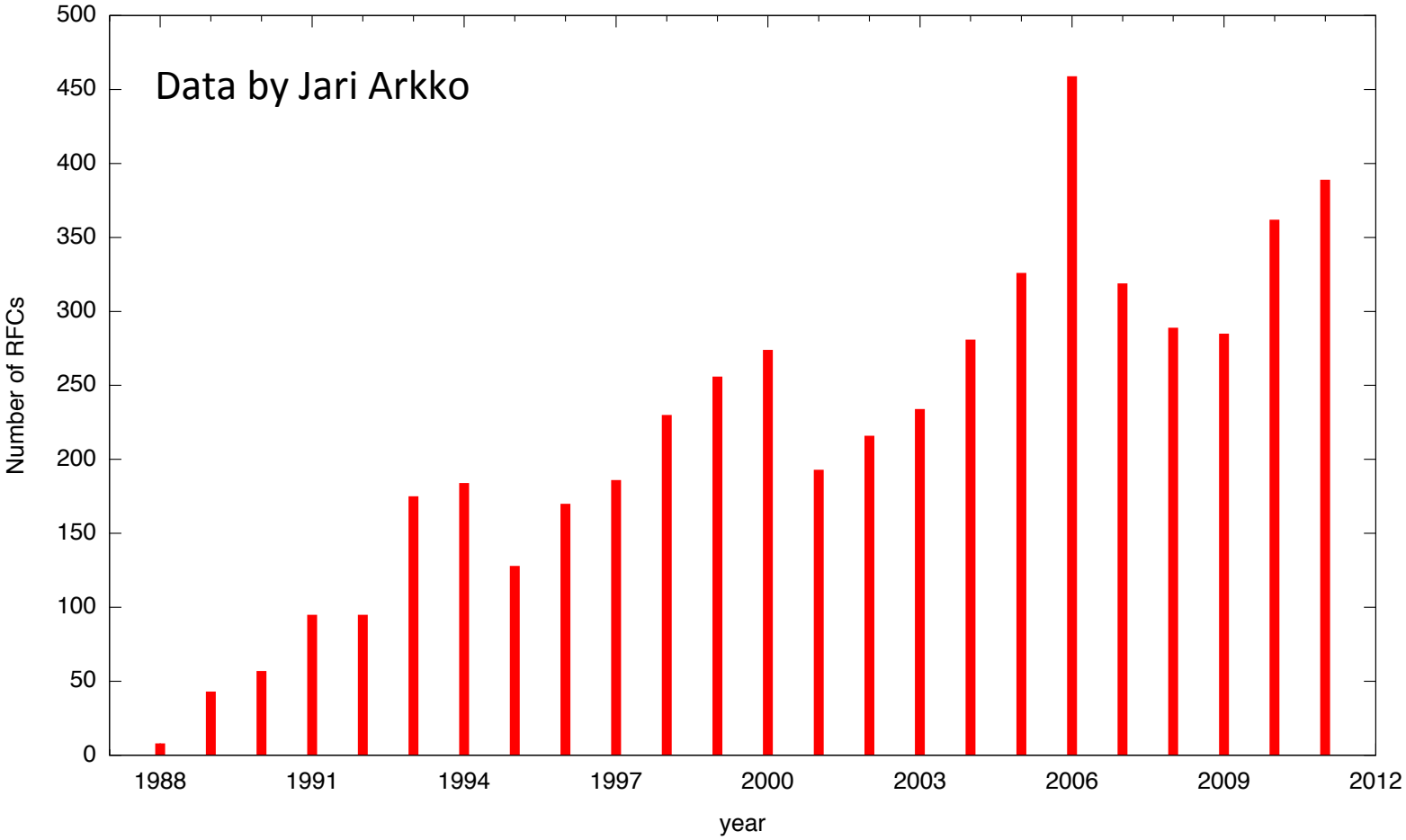
OPEX with OpenFlow

- Run networks similar to computing grids and clouds
- Individual CLI configuration moved to centralised OpenFlow controller configuration
- Application defines policy, translates it to forwarding entries which are sent via the OpenFlow protocol to the OpenFlow switches

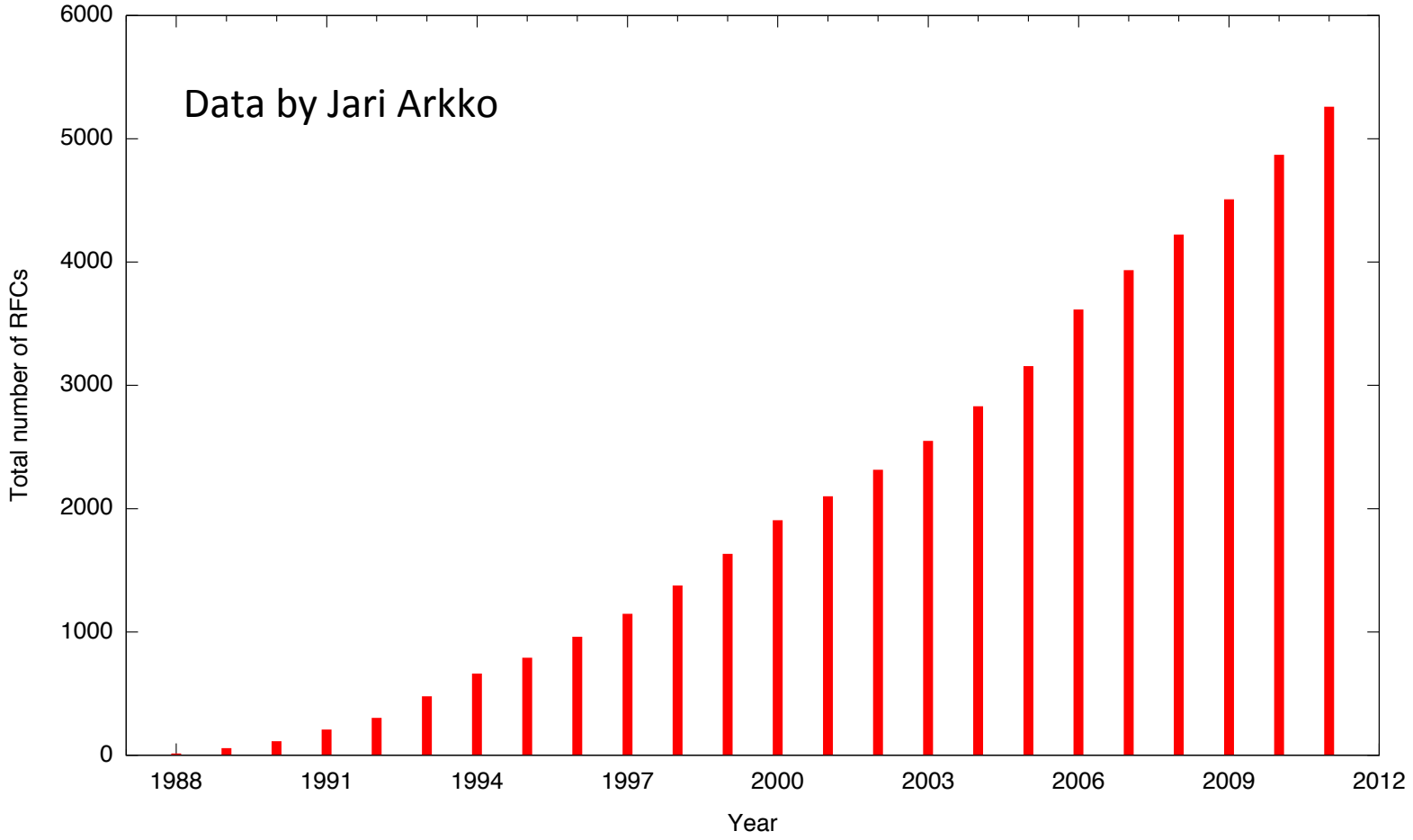
“Protocol Soup”

- 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?

IETF RFC Publication Rate



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)

Number of Supported Protocols in a Modern Ethernet Switch

(random example, but they are all the same)

(STP and RSTP)
- IEEE 802.1w – 2001 Rapid Reconfiguration for STP, RSTP
- IEEE 802.1Q – 2003 (formerly IEEE 802.1s) Multiple Instances of STP, MSTP
- EMISTP, Extreme Multiple Instances of Spanning Tree Protocol
- PVST+, Per VLAN STP (802.1Q interoperable)
- Draft-ietf-bridge-rstpmib-03.txt – Definitions of Managed Objects for Bridges with Rapid Spanning Tree Protocol
- Extreme Standby Router Protocol™ (ESRP)
- IEEE 802.1Q – 1998 Virtual Bridged Local 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
- RFC 1573 Evolution of Interface
- RFC 1650 Ethernet-Like MIB (update of RFC 1213 for SNMPv2)
- RFC 1901, 1905 – 1908 SNMPv2c, SMIv2 and Revised MIB-II
- RFC 2576 Coexistence between SNMP Version 1, Version 2 and Version 3
- RFC 2578 – 2580 SMIv2 (update to RFC 1902 – 1903)
- RFC 3410 – 3415 SNMPv3, user based security, encryption and authentication
- RFC 3826 – The Advanced Encryption

- IEEE 802.1ag MIB
- Secure Shell (SSH-2) client and server
- Secure Copy (SCP-2) client and server
- Secure FTP (SFTP) server
- sFlow version 5
- Configuration logging
- Multiple Images, Multiple Configs
- RFC 3164 BSD Syslog Protocol with Multiple Syslog Servers
  - 999 Local Messages (criticals stored across reboots)
- Extreme Networks vendor MIBs (includes FDB, PoE, CPU, Memory MIBs)
- 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 Copy (SCP-2) and SFTP client/server with encryption/ authentication (requires export controlled encryption module)
- SNMPv3 user based security, with encryption/authentication (see above)
- RFC 1492 TACACS+
- RFC 2138 RADIUS Authentication
- RFC 2139 RADIUS Accounting
- RFC 3579 RADIUS EAP support for 802.1x
- RADIUS Per-command Authentication
- Access Profiles on All Routing Protocols
- Access Policies for Telnet/SSH-2/SCP-2
- Network Login – 802.1x, Web and MAC-based mechanisms
- IEEE 802.1x – 2001 Port-Based Network Access Control for Network Login
- Multiple supplicants with multiple VLANs for Network Login (all modes)
- Fallback to local authentication database (MAC and Web-based methods)
- Guest VLAN for 802.1x
- RFC 1866 HTML – Used for Web-based Network Login and ExtremeXOS ScreenPlay
- SSL/TLS transport – used for Web-based Network Login and ExtremeXOS ScreenPlay (requires export controlled encryption module)
- MAC Security – Lockdown and Limit
- IP Security – RFC 3046 DHCP Option 82 with port and VLAN ID
- IP Security – Trusted DHCP Server
- Layer 2/3/4 Access Control Lists (ACLs)
- RFC 2267 Network Ingress Filtering
- RPF (Unicast Reverse Path Forwarding)
  - CA-97.28:Teardrop_Land -Teardrop and "LAND" attack
  - CA-96.26: ping
  - CA-96.21: tcp_syn_flooding
  - CA-96.01: UDP_service_denial
  - CA-95.01: IP_Spoofing_Attacks_and_ Hijacked_ Terminal_Connections
  - IP Options Attack
- Host Attack Protection
  - Teardrop, boink, opentear, jolt2, newtear, nestea, syndrop, smurf, fraggle, papasmurf, synk4, raped, winfreeze, ping –f, ping of death, pepsi5, Latierra, Winnuke, Simping, Sping, Ascend, Stream, Land, Octopus

Security, Router Protection

- IP Security – DHCP enforcement via Disable ARP Learning
- IP Security – Gratuitous ARP Protection
- IP Security – DHCP Secured ARP/ARP Validation
- Routing protocol MD5 authentication

Security Detection and Protection

- CLEAR-Flow, threshold based alerts and actions

IPv4 Host Services

- RFC 1122 Host Requirements
- RFC 768 UDP
- RFC 791 IP
- RFC 792 ICMP
- RFC 793 TCP
- RFC 826 ARP
- RFC 894 IP over Ethernet
- RFC 1027 Proxy ARP
- RFC 2068 HTTP server
- IGMP v1/v2/v3 Snooping with Configurable Router Registration Forwarding
- IGMP Filters
- PIM Snooping
- Static IGMP Membership
- Multicast VLAN Registration (MVR)

IPv4 Router Services

- RFC 1812 Requirements for IP Version 4 Routers
- RFC 1519 CIDR
- RFC 1256 IPv4 ICMP Router Discovery (IRDP)
- Static Unicast Routes
- Static Multicast Routes
- RFC 1058 RIP v1
- RFC 2453 RIP v2
- Static ECMP
- RFC 1112 IGMP v1

- 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

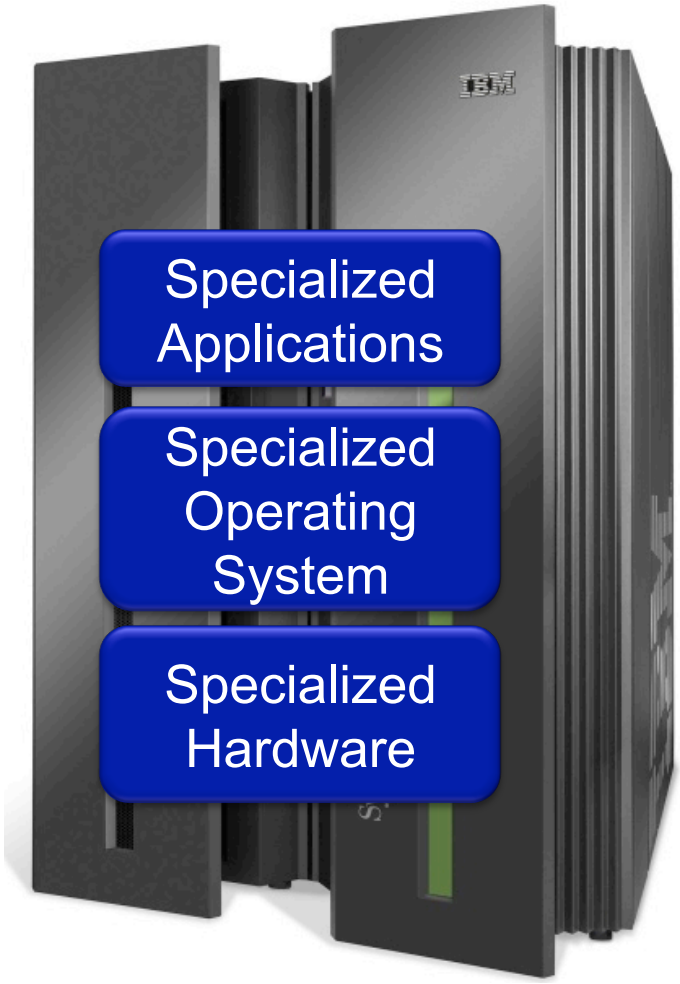
Advanced VLAN Services, MAC-in-MAC
- VLAN Translation in vMAN environments
- vMAN Translation
- IEEE 802.1ah/D1.2 Provider Backbone Bridges (PBB)/MAC-in-MAC

MPLS and VPN Services

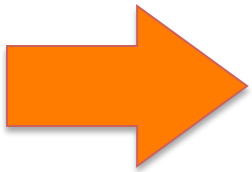
Multi-Protocol Label Switching (MPLS)
Requires MPLS Layer 2 Feature Pack License
- RFC 2961 RSVP Refresh Overhead Reduction Extensions
- RFC 3031 Multiprotocol Label Switching Architecture
- RFC 3032 MPLS Label Stack Encoding
- RFC 3036 Label Distribution Protocol (LDP)
- RFC 3209 RSVP-TE: Extensions 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

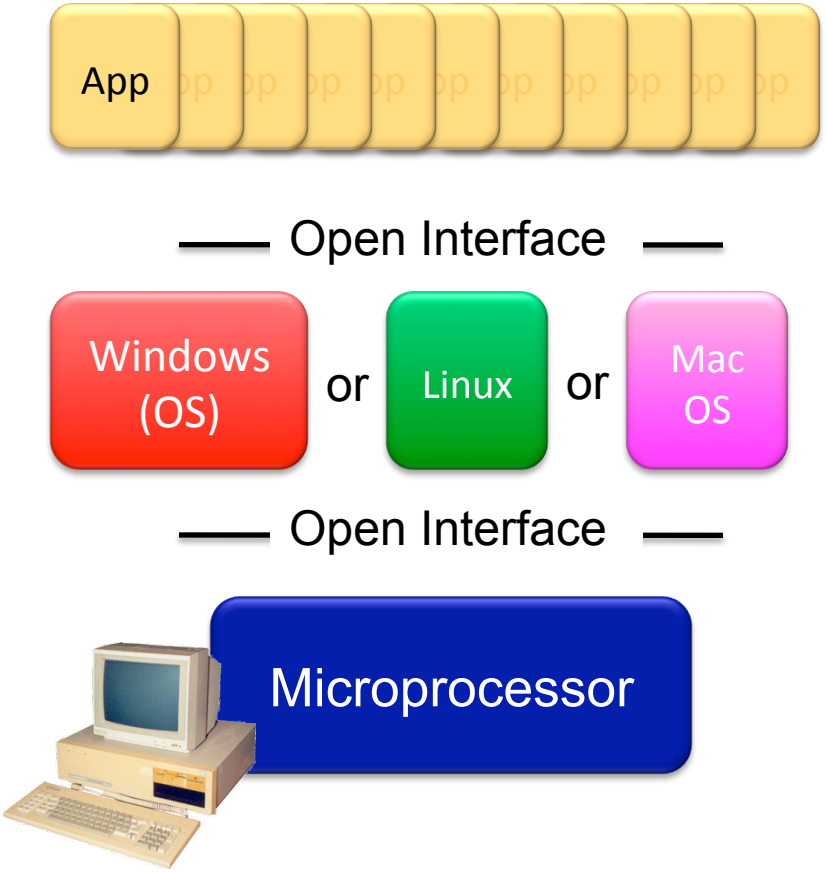
Layer 2 VPNs
Requires MPLS Layer 2 Feature Pack License
- RFC 4447 Pseudowire Setup and Maintenance using the Label Distribution Protocol (LDP)
- RFC 4448 Encapsulation Methods for Transport of Ethernet over MPLS Networks
- RFC 4762 Virtual Private LAN Services (VPLS) using Label Distribution Protocol (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

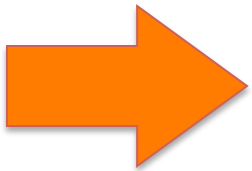
(slide by Nick McKeown, Stanford University)



Vertically integrated
Closed, proprietary
Slow innovation
Small industry

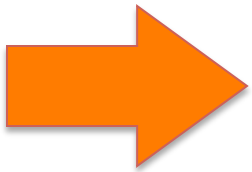


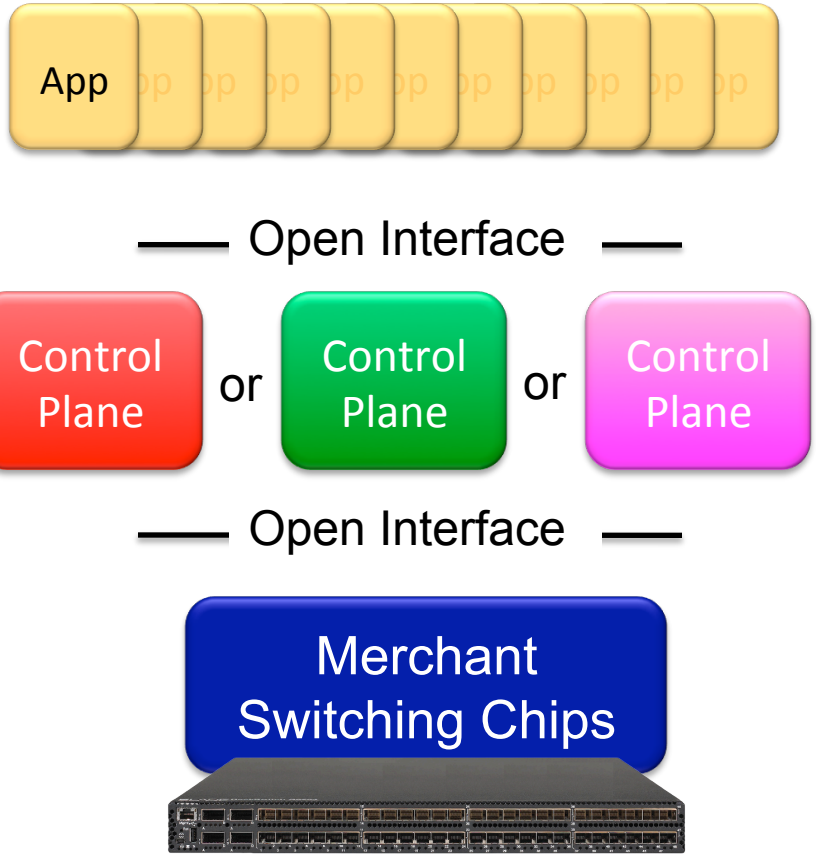




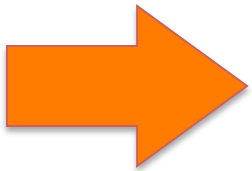
Horizontal
Open interfaces
Rapid innovation
Huge industry

(slide by Nick McKeown, Stanford University)





Vertically integrated
Closed, proprietary
Slow innovation

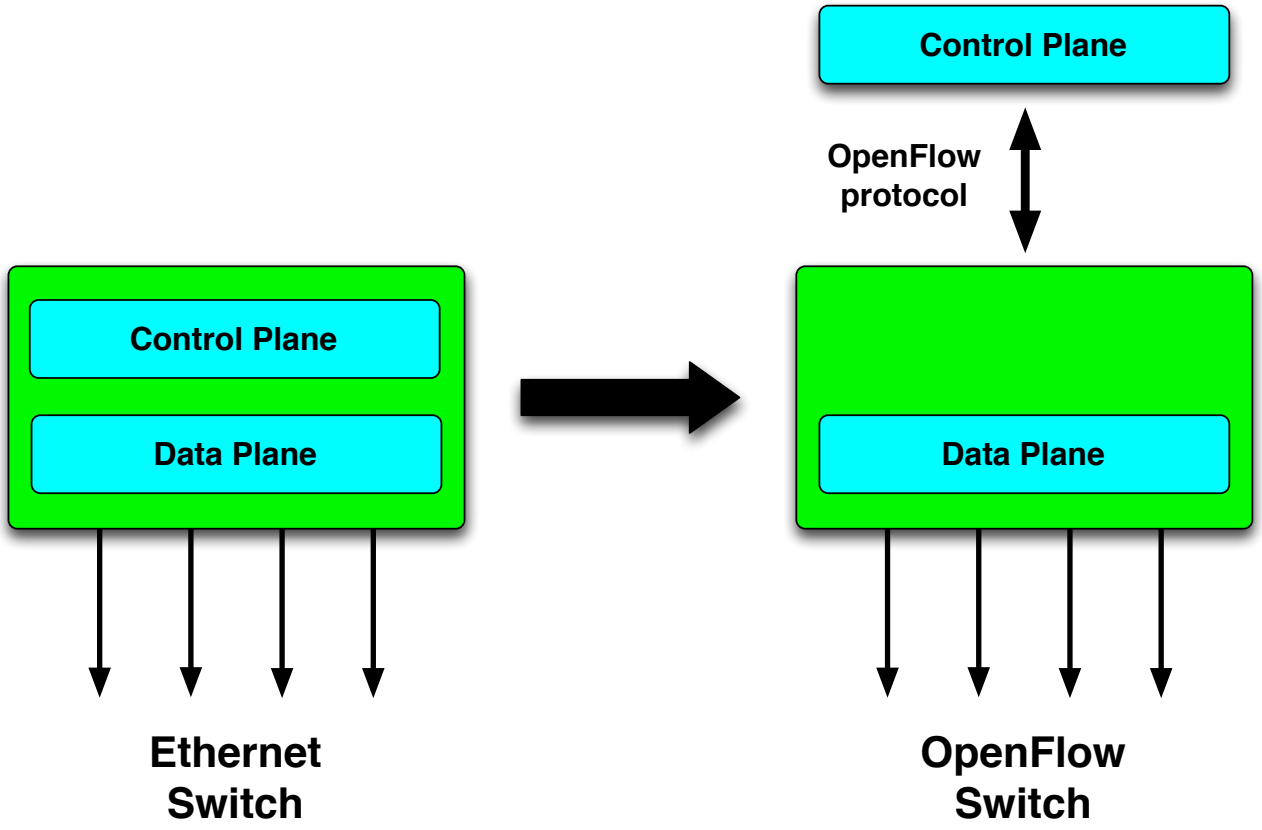


Horizontal
Open interfaces
Rapid innovation

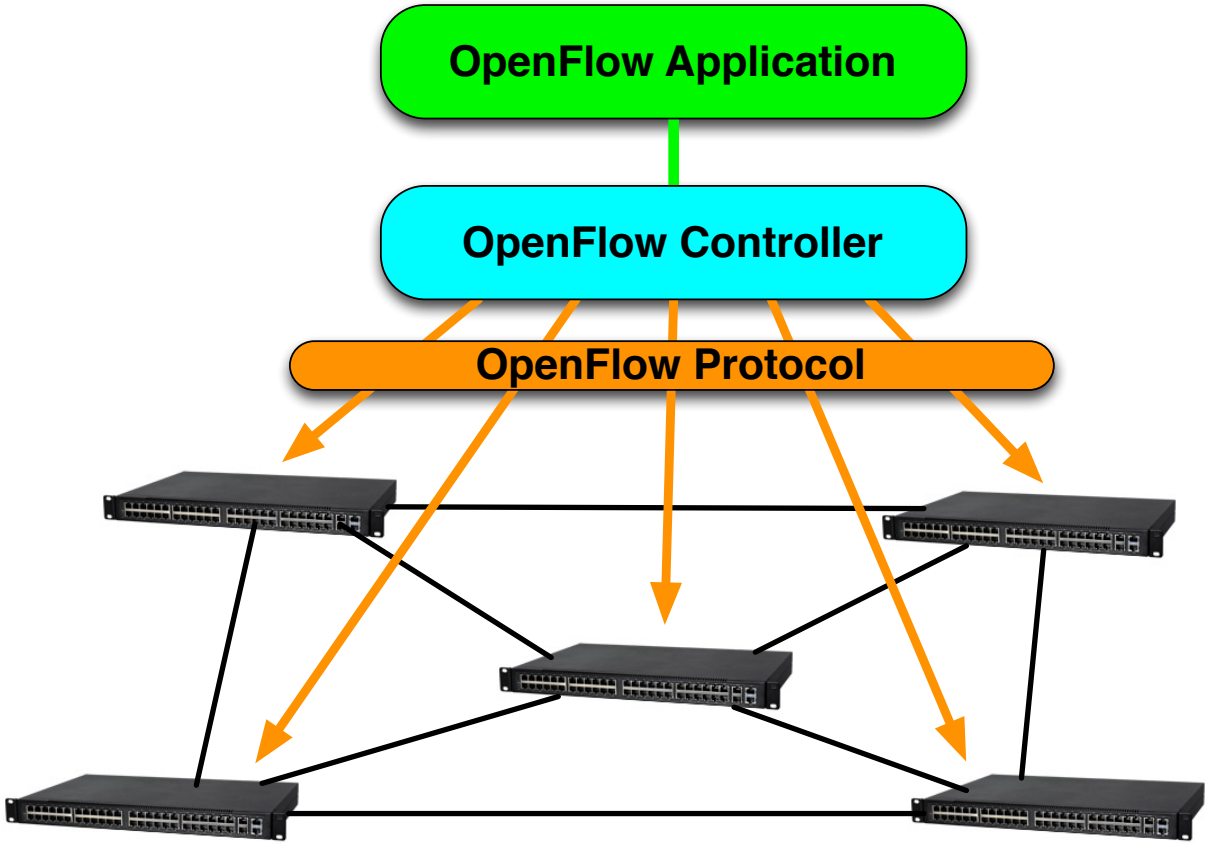
How Does OpenFlow Work?

- Control Plane moved out of the switch
- Standardised protocol between Data Plane and Control Plane → OpenFlow
- OpenFlow controller typically connects to many switches
- Centralised view of the whole network
- Centralised does not mean a SPOF; The SDN application can run on a (distributed) computer cluster

Data and Control Plane Separation



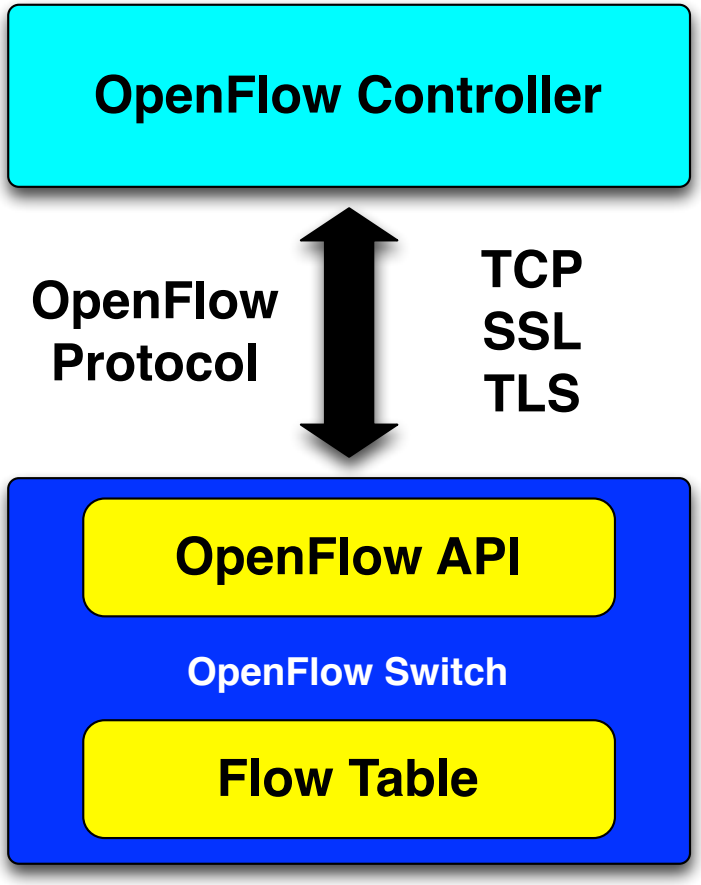
OpenFlow Controlled Network



OpenFlow Protocol

- Insert flow forwarding entries in switches
- Send packets to OpenFlow switch
- Receive packets from OpenFlow switch
- Receive traffic statistics from OpenFlow switch
- Retrieve flow tables from OpenFlow switch
- Retrieve parameters from OpenFlow switch
  - E.g. number of ports

OpenFlow Components



Flow Table

- *Header Fields*: match against packets
- *Counters*: count matching packets
- *Actions*: Actions to take when packet matches

Header Matching (OF 1.0)

- Ingress port
- Ethernet source/destination address
- Ethernet type
- VLAN ID
- VLAN priority
- IPv4 source/destination address
- IPv4 protocol number
- IPv4 type of service
- TCP/UDP source/destination port
- ICMP type/code

Counters (1/3)

- Per table:
  - Active entries (32 bits)
  - Packet lookups (64bits)
  - Packet matches (64 bits)
- Per flow:
  - Received packets (64 bits)
  - Received bytes (64 bits)
  - Duration <seconds> (32 bits)
  - Duration <nanoseconds> (32 bits)

Counters (2/3)

- Per port:
  - Received/Transmitted packets (64 bits)
  - Received/Transmitted bytes (64 bits)
  - Receive/Transmit drops (64 bits)
  - Receive/Transmit errors (64 bits)
  - Receive frame alignment errors (64 bits)
  - Receive overrun errors (64 bits)
  - Receive CRC errors (64 bits)
  - Collisions

Counters (3/3)

- Per queue:
  - Transmit packets (64 bits)
  - Transmit bytes (64 bits)
  - Transmit overrun errors (64 bits)

Actions

- Forward
  - Required: All, Controller, Local, Table, IN_PORT
  - Optional: Normal, Flood
- Enqueue (Optional)
- Drop (Required)
- Modify Field (Optional)

Required Forward Actions

- All
  - Sent packet out on all interfaces, not including incoming interface
- Controller
  - Encapsulate and send the packet to the controller
- Local
  - Send the packet to the switch local network stack
- Table
  - Perform action in flow table (for packet_out)
- IN_PORT
  - Send the packet out to the input port

Optional Forward Actions

- Normal
  - Process the packet using the traditional forwarding path supported by the switch
- Flood
  - Flood the packet along the spanning tree, not including the incoming interface

Optional Modify Field Action

- Set VLAN ID (or add VLAN tag)
- Set VLAN priority
- Strip VLAN header
- Modify Ethernet source/destination address
- Modify IPv4 source/destination address
- Modify IPv4 type of service bits
- Modify TCP/UDP source/destination 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=output:6,mod_vlan_id:42
  - ovs-ofctl add-flow br0 in_port=6,actions=output:4,strip_vlan

OpenFlow Standardisation

- Open Networking Foundation (ONF)
- Non-Profit consortium
- Founded in March 2011 by Deutsche Telecom, Facebook, Google, Microsoft, Verizon and Yahoo!
- Mission: promotion of Software Defined Networking (SDN)

OpenFlow Switch Specification

- Defines OpenFlow protocol
- OpenFlow Switch Specification 1.0.0 (Dec 31, 2009)
  - Most widely used version today
- OpenFlow Switch Specification 1.1.0 (Feb 28, 2011)
  - QinQ and MPLS support
  - Skipped by most vendors
- OpenFlow Switch Specification 1.2 (Dec 2011)
  - IPv6 support, extensible matches
- OpenFlow 1.3.0 (Jun 25, 2012)
  - Flexible table miss, per flow meters, PBB support
- OpenFlow 1.3.1 (Sep 6, 2012)

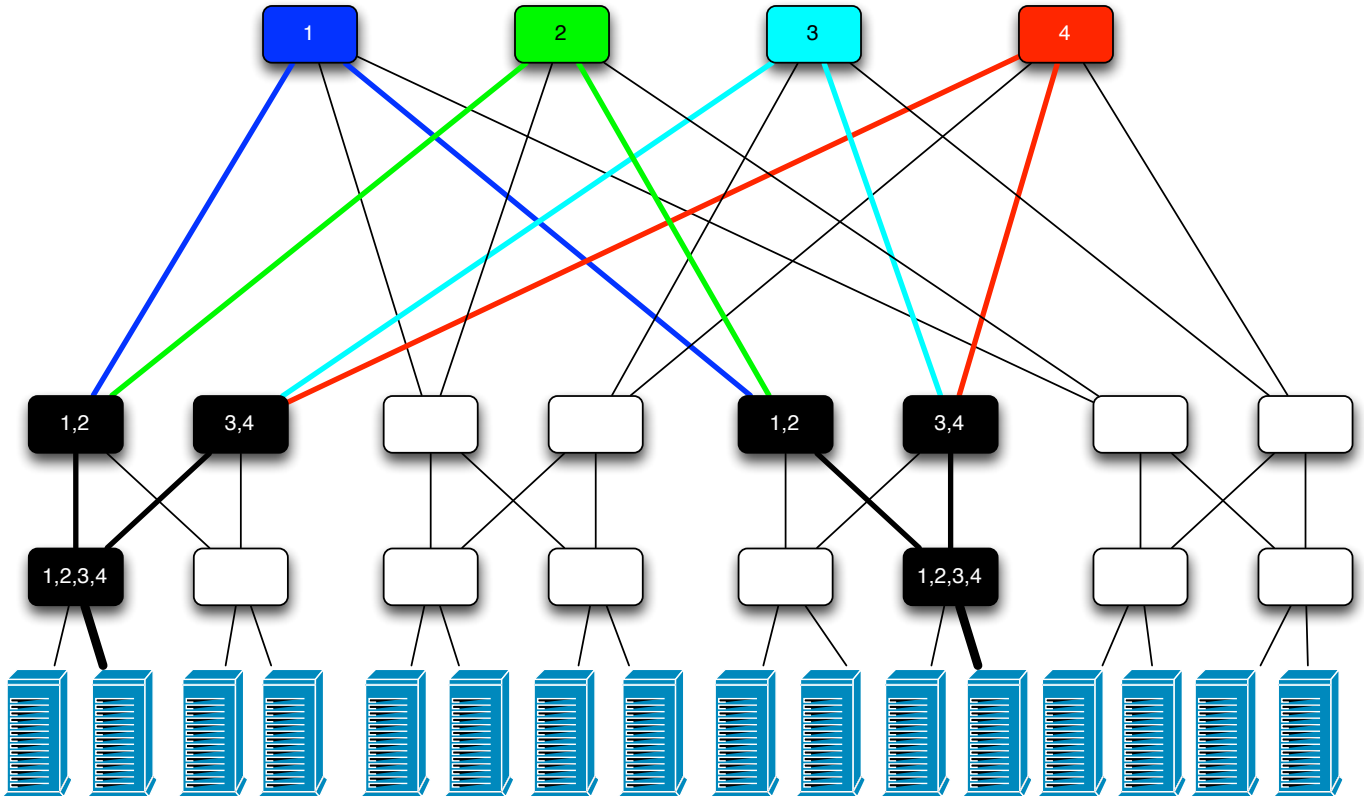
OF-Config

- Configuration protocol for setting up ports, controller IP address, etc.
- Netconf based
- OF-Config 1.0 (Sep 2012)
- OF-Config 1.1 (Jun 25, 2012)

OpenFlow in Data Centres

- Cloud middleware (OpenStack, etc) handles compute & storage resources (VMs, disks)
- Now also including network resources (OpenStack Quantum)
- SDN and OpenFlow perfect match in this ecosystem

Fat Tree Data Centre Network



Insert flow entries to use multiple path through network
Support multiple virtual networks (multiple tenants)
Adjust flow entries when VMs migrate (move network with VMs)

Fat Tree Topologies

- Three stage fat tree with k-port switches
  - Non-blocking forwarding between k^3/4 servers
  - 5K^2/4 switches need with k ports
- E.g. 4-port switches:
  - 16 servers
  - 20 switches
- 128-port switches:
  - 524,288 servers
  - 20,480 switches

Forwarding in Campus Networks

- VLAN configuration and maintenance complex and error prone
- Spanning Tree Protocol (STP) results in sub-optimal link usage
- Link usage difficult to control with STP
- Related data centre fabric technologies:
  - Cisco FabricPath
  - Juniper Qfabric
  - IETF TRILL
  - Brocade Virtual Cluster Switching (VCS)
  - IEEE 802.1aq Shortest Path Bridging (SPB)
  - NVGRE, VXLAN

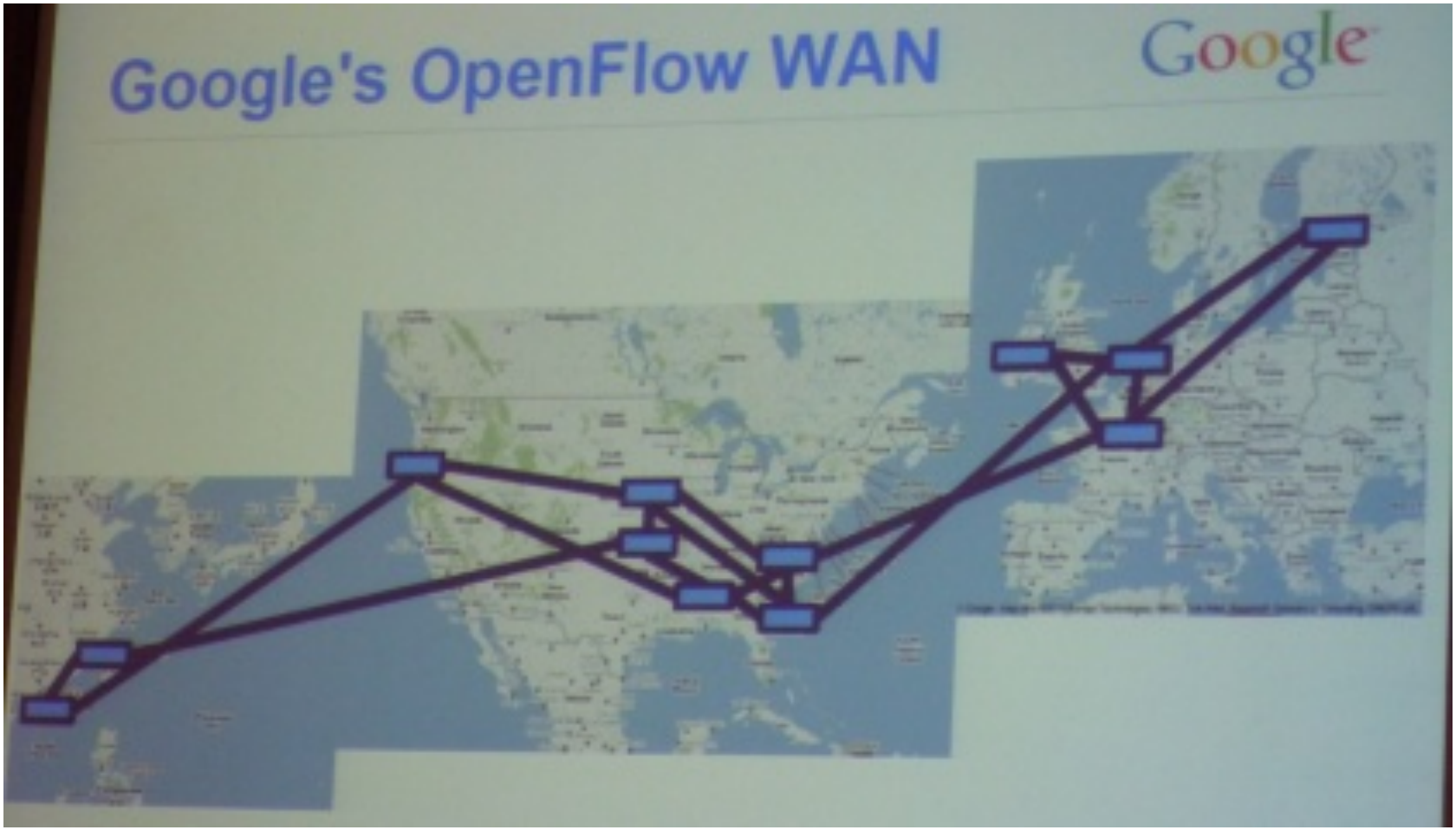
Security in Campus Networks

- Complex and difficult to maintain firewalling
  - Hard to get an overview of what traffic is allowed/blocked
  - Difficult to get consistent policy when multiple ACL/firewalls are used
- Traffic is sometimes forwarded inside campus and dropped at faculty edge
- SDN/OpenFlow has several advantages
  - Only forward allowed traffic
  - Centralised policy that is compiled and pushed to the switches
  - Central SDN application can be combined with other policy databases (Active Directory, LDAP, radius, user database, etc)

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
- Google uses custom built switches with merchant chip sets (128 ports of 10GE)
  - Custom build just because such switches were not commercially available yet
  - Next (commercial) switch will probably have 1000+ ports of 40GE (2013)

Google Data Network



Slide by Google

Google Data Network

- Goal:
  - Improve backbone performance
  - Reduce complexity and cost
  - Cost per bit/s should go down when scaling up
    - Today there is a quadratic increase (N nodes talking to each other)
    - Configuration cost of adding a node
    - Broadcast traffic required more expensive hardware
  - Control Plane on commodity hardware
  - Faster and better TE decisions
    - TE decisions with global knowledge about network instead of local knowledge

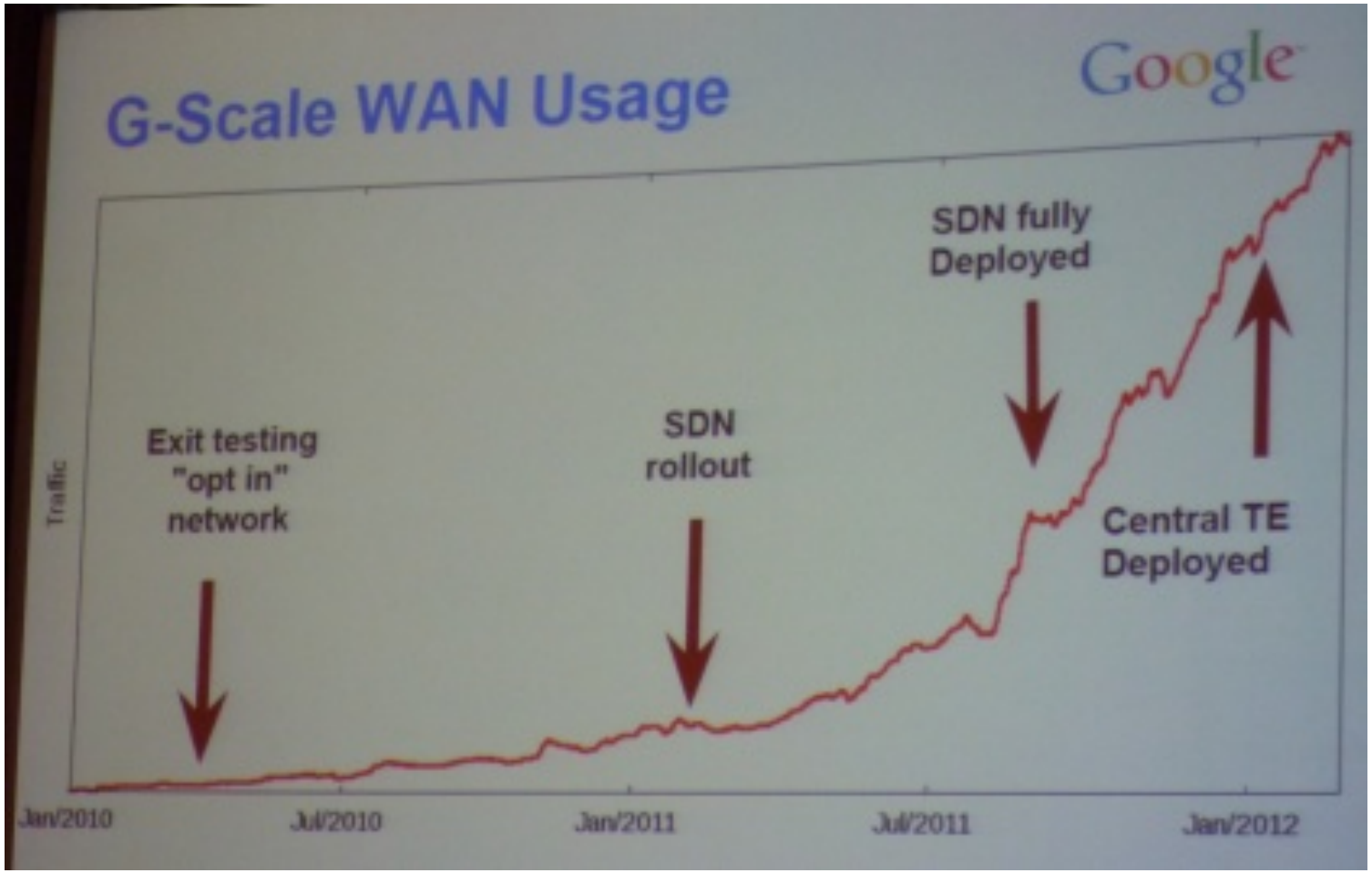
Google Data Network

- Issues with current equipment
  - Internet protocols are box centric, not fabric centric
  - Monitoring added as an afterthought

Google Data Network

- Multiple controllers
  - 3, 5, 7 with Paxos majority voting (my assumption)
- The whole network can be 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 Data Network



Slide by Google

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)

Some OpenFlow Players

- Startups
  - Big Switch Networks
  - Nicira (aquired by VMware for 1 Billion USD in 2012)
  - Pica8
- ONRC
- ON.LAB

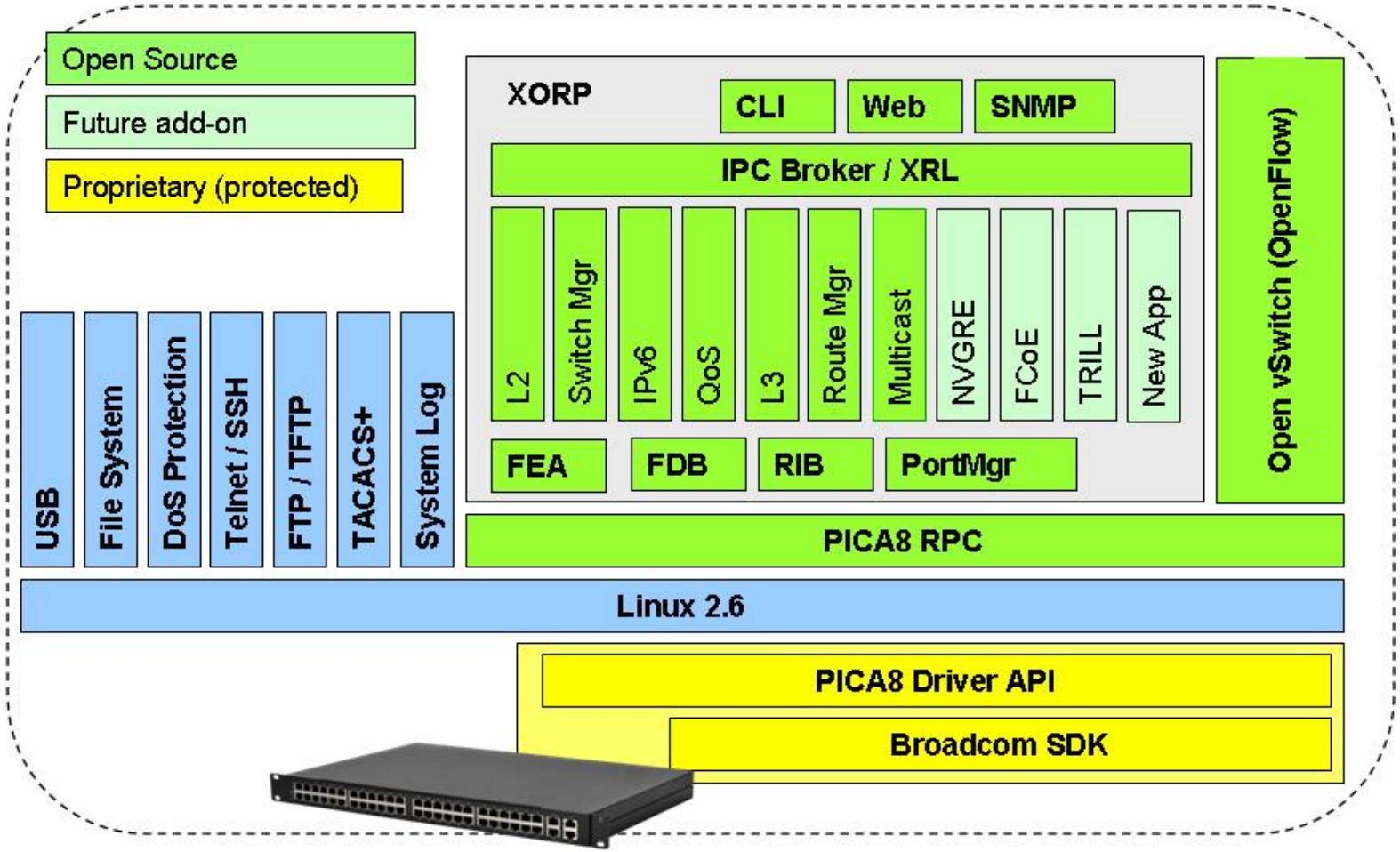
Stanford University Startups

- Big Switch Networks
  - FloodLight (open source OpenFlow controller)
  - Guido Appenzeller (CEO/Co-Founder)
    - Former head of Stanford University Clean Slate program
  - Kyle Forster (Co-Founder, ex-Cisco, ex-Joost)
- Nicira
  - Open vSwitch (open source software switch)
  - Steve Mullaney (CEO)
  - Martin Casado (CTO, Co-Founder)
    - SDN was graduate work at Stanford University, supervised by Nick KcKeown & Scott Shenker
  - Nick McKeown (Co-Founder)
    - Former faculty director of Stanford University Clean Slate program
  - Scott Shenker (Chief Scientest, Co-Founder)
    - University of California at Berkeley

Pica8

- Founded in 2008
- Open the switch and router platforms
- High quality software with commoditised switches
- PicOS based on:
  - XORP (open source routing project)
  - Open vSwitch (open source OpenFlow switch)

PicOS Architecture



Pica8 (Pronto) Switches

- Pica8 3290
  - 48x 10/100/1000 BASE-T RJ45 & 4x 10GE SFP+
  - USD 2,750
- Pica8 3780
  - 48x 10GE SFP+
  - USD 9,950
- Pica8 3920
  - 48x 10GE SFP+ & 4x 40GE QSFP
  - USD 13,500

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

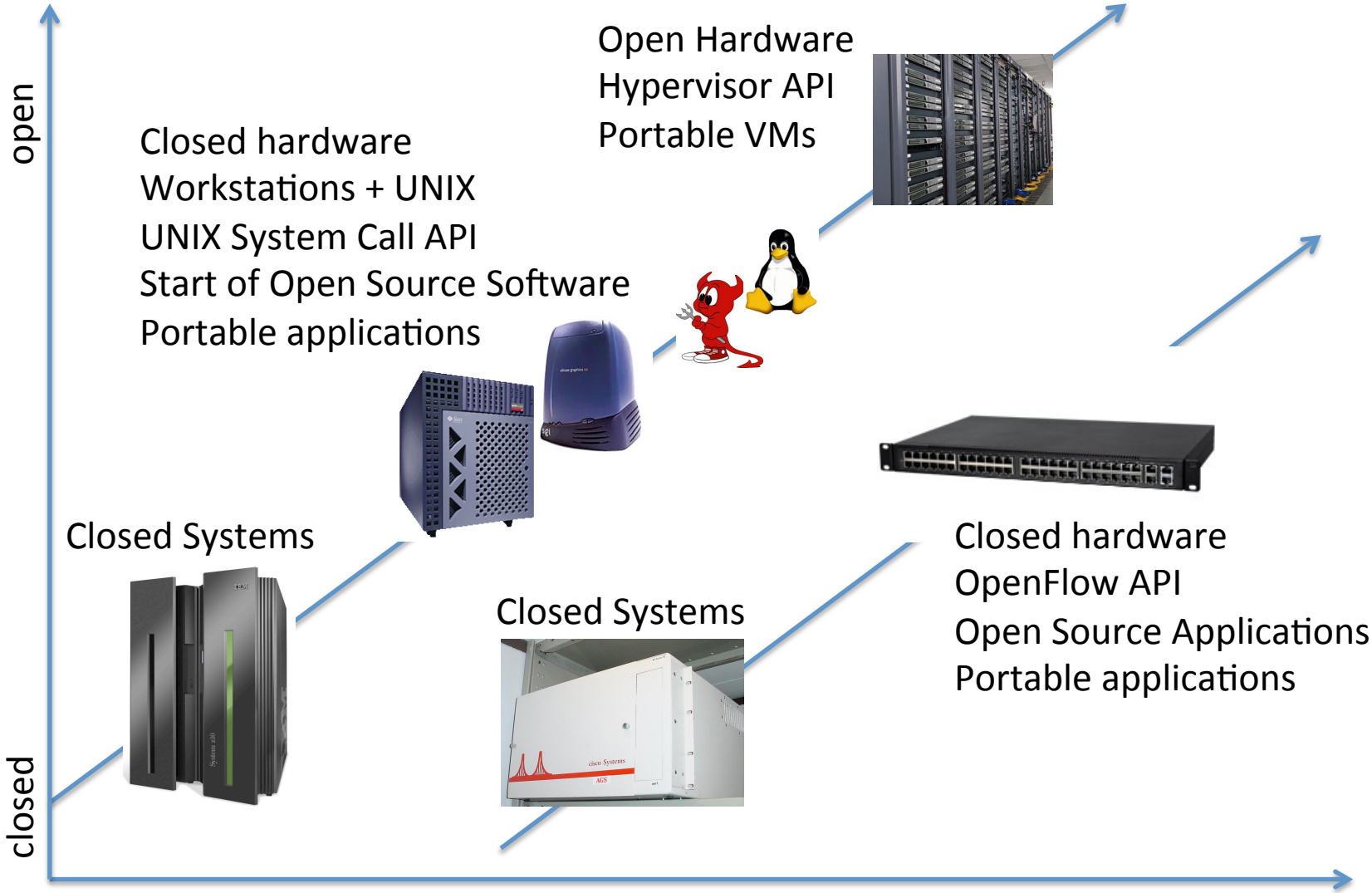
ON.LAB

- Headed by Guru Parulkar
  - Professor at Stanford University
- Build open source OpenFlow tools and platforms
  - Beacon, NOX, FlowVisor, Mininet

Conclusions

- OpenFlow has got a lot of attention in 2011/2012
- Centralised programmable control makes network management easier (write policy instead of detailed switch/router configurations)
- *Possible* disruptive (network) technology (time will tell)
- Very likely it will be used within data centres combined with cloud middleware
- Could be the start of an open hardware/open software network ecosystem

Computing vs Networking



Thank You

Ronald.vanderPol@SURFnet.nl