

# OpenFlow

- 



## Member Listing

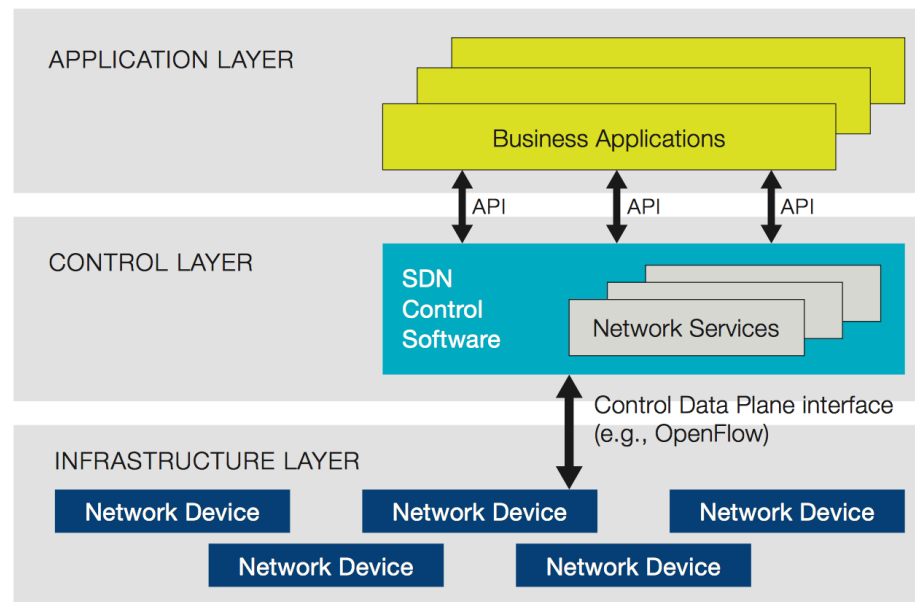


## STARTUP MEMBERS



# ONF y SDN

- *“The aim of SDN is to provide open interfaces that enable the development of software that can control the connectivity provided by a set of network resources and the flow of network traffic through them, along with possible inspection and modification of traffic that may be performed in the network.”*
- *“In the SDN architecture, the control and data planes are decoupled, network intelligence and state are logically centralized, and the underlying network infrastructure is abstracted from the applications.”*



# OpenFlow

- Dos tipos de conmutadores:
  - *OpenFlow-only*: solo soportan el modo de funcionamiento OpenFlow
  - *OpenFlow-hybrid*: también soportan funcionamiento “normal” (conmutación L2, conmutación L3, VLANs, ACLs, etc)
  - Los híbridos deberán tener alguna forma de clasificar si los paquetes pasan por procesado “normal” u OpenFlow



# Ejemplo: HP 2920-24G

## Key Features

- High-performance Gigabit Ethernet access switch
- Four optional 10GbE (SFP+ and/or 10GBASE-T) ports
- Stacking capability with a total of four switches
- L2 and L3 plus static and RIP routing, PoE, and PoE+ support
- Limited Lifetime Warranty 2.0, sFlow, ACLs, OpenFlow, and rate limiting



## Product overview

The HP 2920 Switch Series consists of five switches: the 2920-24G and 2920-24G-PoE+ switches with 24 10/100/1000 ports and the 2920-48G, 2920-48G-PoE+, and 2920-48G 740W PoE+ switches with 48 10/100/1000 ports. Each switch has four dual-personality ports for 10/100/1000 or SFP connectivity.

In addition, the 2920 Switch Series supports up to four optional 10 Gigabit Ethernet (SFP+ and/ or 10GBASE-T) ports, as well as a two-port stacking module. These options provide you with flexible and easy-to-deploy uplinks and stacking.

Together with static and routing-information-protocol (RIP) routing, robust security and management, enterprise-class features, Limited Lifetime Warranty 2.0, and software updates included, the 2920 Switch Series is a comprehensive, cost-effective, and scalable solution for building high-performance networks. These switches can be deployed at the enterprise edge, in remote branch offices, and in converged networks.

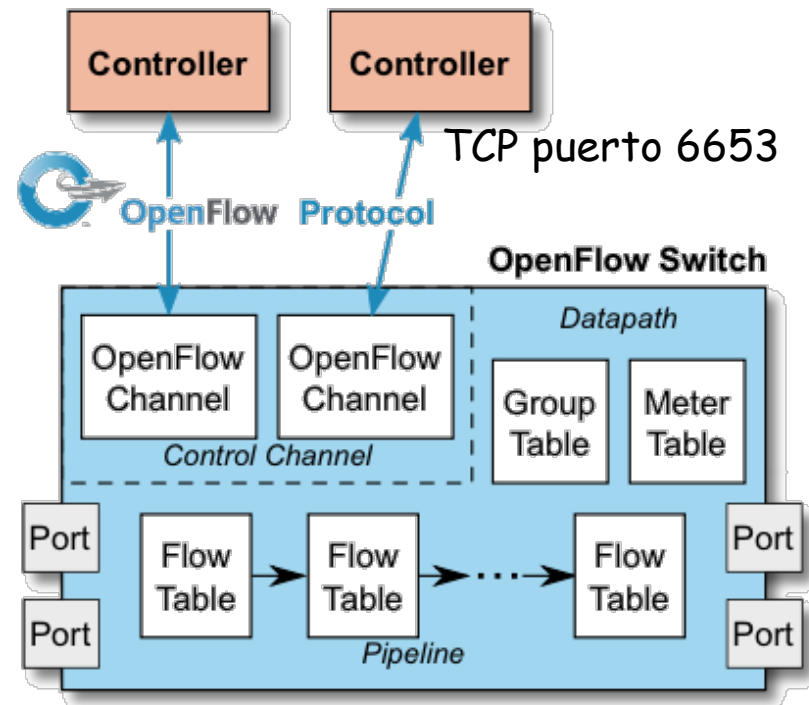
## Features and Benefits

### Software-defined networking

- **OpenFlow**  
supports OpenFlow 1.0 and 1.3 specifications to enable SDN by allowing separation of the data (packet forwarding) and control (routing decision) paths

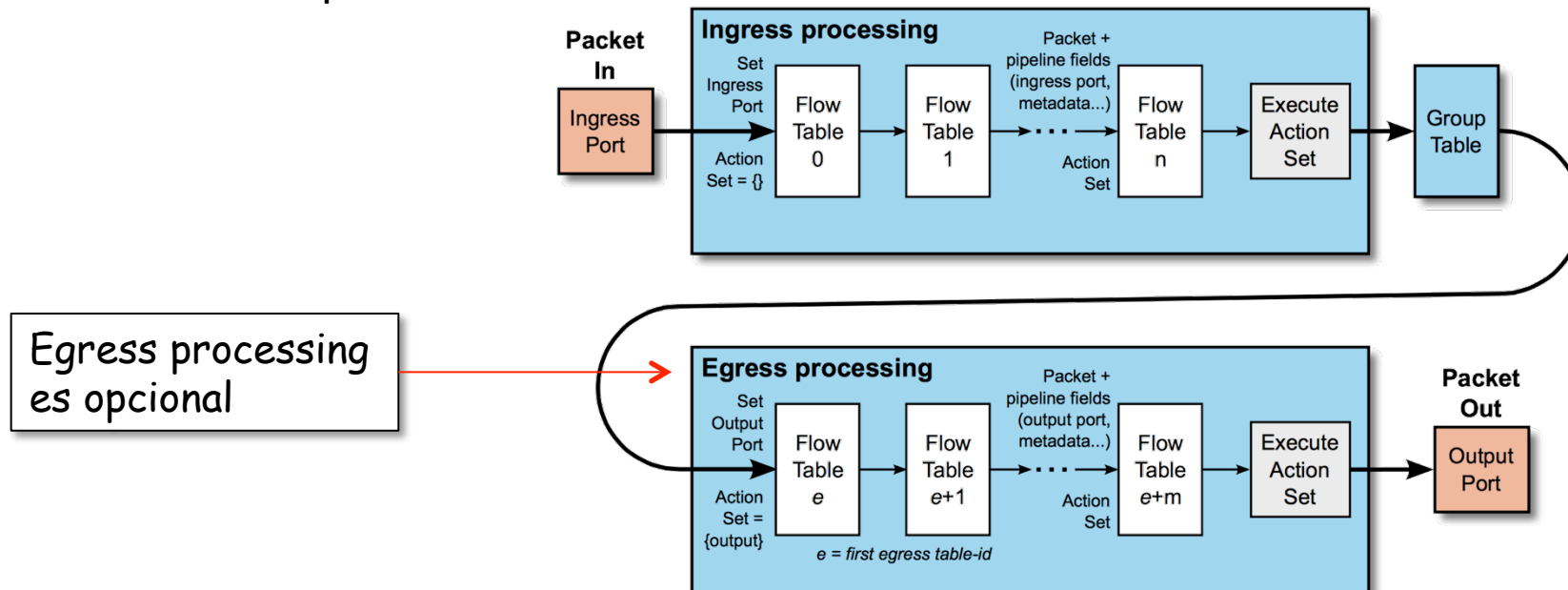
# Flow Tables

- Contienen la información sobre los campos a comprobar (*match fields*) en los paquetes y qué hacer con ellos
- El controlador puede añadir, modificar y borrar entradas empleando OF
- Las “acciones” son las operaciones en caso de que el paquete verifique la entrada en la tabla
- Puede reenviar el paquete, mandárselo al controlador, pasarlo a otra tabla, actualizar contadores, etc



# OpenFlow pipeline

- Debe tener al menos una tabla aunque pueden ser más (desde 1.1, permite procesamiento de etiquetas MPLS)
- Hay procesamiento a la entrada del paquete (al menos una tabla)
- Si se decide reenviarlo pasa por tablas de salida (desde 1.5)
- Las tablas se comprueban en orden
- Si el paquete verifica una regla se ejecuta la acción que indique
- Si no verifica ninguna es un *"table miss"* y hay una acción por defecto en la tabla para este caso



# Acciones

- Incluimos aquí la acción por defecto para el caso de “*table miss*”
- La acción puede ser pasar a otra tabla posterior (no anterior)
- Puede ser hacer inundación
- O reenviar por un puerto en concreto
- O puede ser reenviar el paquete al controlador (dentro de un mensaje OF)
- O pasar el paquete a un reenvío tradicional si es un conmutador híbrido
- O modificar campos de cabeceras del paquete (una modificación afecta a las comprobaciones en egress tables)
- etc

# Entradas en las tablas

- *Match Fields:*
  - Puede valer ANY (comodín) o soportarse bitmasks
  - Hasta la versión 1.1 se miraban ciertos campos:
    - Puerto de entrada, metadatos provenientes de tabla anterior
    - Direcciones MAC origen y destino, Ethertype, VLAN ID, PCP
    - Etiqueta MPLS, TC
    - Direcciones IP origen y destino, protocolo, ToS
    - Puertos origen y destino TCP/UDP/SCTP
    - Tipo y código ICMP
  - Otros que se han ido añadiendo:
    - Bits ECN
    - Flags TCP
    - Código de opción de ARP, direcciones MAC e IP en el mensaje ARP
    - Direcciones IPv6, flow label IPv6, tipo y código ICMPv6
    - Etc
  - (...)

Match Fields	Priority	Counters	Instructions	Timeouts	Cookie	Flags

# Entradas en las tablas

- Prioridad:
  - Pueden verificarse varias entradas de la tabla
  - En ese caso se selecciona solo la de mayor prioridad
- (...)

Match Fields	Priority	Counters	Instructions	Timeouts	Cookie	Flags

# Entradas en las tablas

- Contadores:
  - Se actualizan cuando la entrada es seleccionada
- (...)

Counter	Bits	
Per Flow Table		
Reference Count (active entries)	32	<i>Required</i>
Packet Lookups	64	<i>Optional</i>
Packet Matches	64	<i>Optional</i>
Per Flow Entry		
Received Packets	64	<i>Optional</i>
Received Bytes	64	<i>Optional</i>
Duration (seconds)	32	<i>Required</i>
Duration (nanoseconds)	32	<i>Optional</i>
Per Port		
Received Packets	64	<i>Required</i>
Transmitted Packets	64	<i>Required</i>
Received Bytes	64	<i>Optional</i>
Transmitted Bytes	64	<i>Optional</i>
Receive Drops	64	<i>Optional</i>
Transmit Drops	64	<i>Optional</i>
Receive Errors	64	<i>Optional</i>
Transmit Errors	64	<i>Optional</i>
Receive Frame Alignment Errors	64	<i>Optional</i>
Receive Overrun Errors	64	<i>Optional</i>
Receive CRC Errors	64	<i>Optional</i>
Collisions	64	<i>Optional</i>
Duration (seconds)	32	<i>Required</i>
Duration (nanoseconds)	32	<i>Optional</i>

Per Queue		
Transmit Packets	64	<i>Required</i>
Transmit Bytes	64	<i>Optional</i>
Transmit Overrun Errors	64	<i>Optional</i>
Duration (seconds)	32	<i>Required</i>
Duration (nanoseconds)	32	<i>Optional</i>
Per Group		
Reference Count (flow entries)	32	<i>Optional</i>
Packet Count	64	<i>Optional</i>
Byte Count	64	<i>Optional</i>
Duration (seconds)	32	<i>Required</i>
Duration (nanoseconds)	32	<i>Optional</i>
Per Group Bucket		
Packet Count	64	<i>Optional</i>
Byte Count	64	<i>Optional</i>
Per Meter		
Flow Count	32	<i>Optional</i>
Input Packet Count	64	<i>Optional</i>
Input Byte Count	64	<i>Optional</i>
Duration (seconds)	32	<i>Required</i>
Duration (nanoseconds)	32	<i>Optional</i>
Per Meter Band		
In Band Packet Count	64	<i>Optional</i>
In Band Byte Count	64	<i>Optional</i>

# Entradas en las tablas

- *Instructions:*
  - Cambio al paquete, acciones, etc, cuando se selecciona la entrada
  - Las hay de implementación requerida y opcional
  - Ejemplos:
    - Enviar a un puerto de salida, descartar, asignar cola en el puerto out
    - Añadir/retirar etiquetas (MPLS, VLAN, PBB)
    - Modificar valor de un campo de cabecera
- (...)

Match Fields	Priority	Counters	Instructions	Timeouts	Cookie	Flags



# Entradas en las tablas

- *Timeouts:*
  - Máximo tiempo inactiva antes de expirar
- (...)

Match Fields	Priority	Counters	Instructions	Timeouts	Cookie	Flags

# Entradas en las tablas

- *Cookie*:
  - Ahí el controlador puede guardar un valor
  - El switch no lo emplea para nada
- (...)

Match Fields	Priority	Counters	Instructions	Timeouts	<i>Cookie</i>	Flags

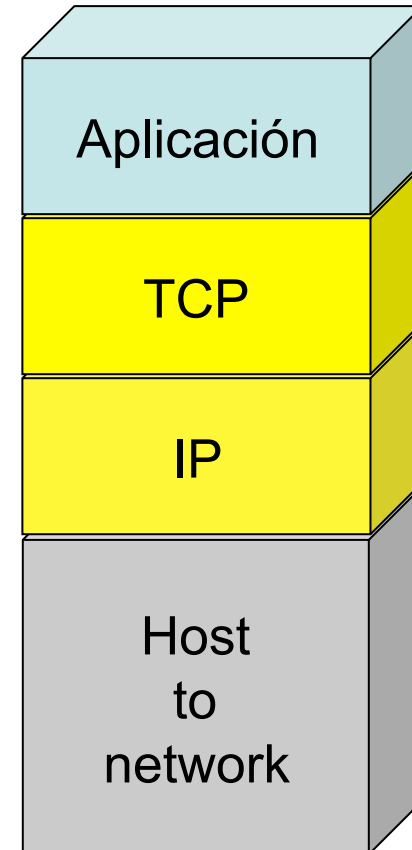
# Entradas en las tablas

- *Flags:*
  - Diferentes opciones
  - Ejemplo:
    - Que envíe un mensaje al controlador al eliminarse o expirar una entrada
    - Que no lleve contadores de bytes o de paquetes

Match Fields	Priority	Counters	Instructions	Timeouts	Cookie	Flags

# El protocolo

- TCP (puerto 6653), opcionalmente empleando TLS
- Hay mensajes:
  - De controlador a conmutador (...)
  - Asíncronos (desde el conmutador)
  - Simétricos



# El protocolo

- TCP (puerto 6653), opcionalmente empleando TLS
- Hay mensajes:
  - De controlador a conmutador
    - Petición de capacidades
    - Establecer o preguntar por configuración o estado
    - Entregarle un paquete para enviar por un puerto
  - Asíncronos (desde el conmutador) (...)
  - Simétricos



# El protocolo

- TCP (puerto 6653), opcionalmente empleando TLS
- Hay mensajes:
  - De controlador a conmutador
  - Asíncronos (desde el conmutador)
    - Envío al controlador de un paquete recibido
    - Notificación de entrada en tabla eliminada
    - Notificación de cambio de estado de un puerto
  - Simétricos (...)



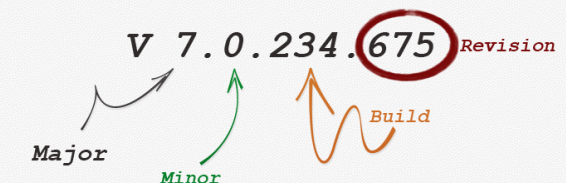
# El protocolo

- TCP (puerto 6653), opcionalmente empleando TLS
- Hay mensajes:
  - De controlador a conmutador
  - Asíncronos (desde el conmutador)
  - Simétricos
    - Hello, al establecer la conexión
    - Echo, para comprobar que el otro extremo está vivo y tal vez para medir latencia o bw
    - Error



# Versiones

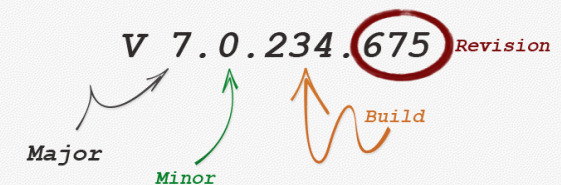
- <https://www.opennetworking.org/sdn-resources/technical-library>
- Versión 1.5.1 Abril de 2015
- Probablemente OF 1.0 sea lo más implementado en hardware
- Las siguientes versiones han ido introduciendo mejoras, más flexibilidad, pero también haciéndolo más complejo
- OF 1.1
  - Múltiples tablas
  - Soporte de acciones para MPLS (soporta multi-etiqueta)
  - Acciones sobre el TTL
  - Soporte de VLANs en QinQ
  - Soporte para agrupar puertos de cara a acciones
- (...)





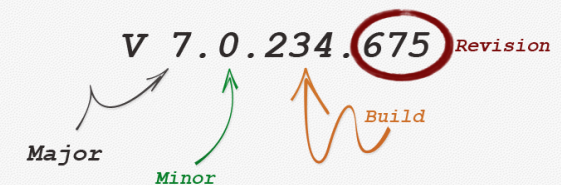
# Versiones

- <https://www.opennetworking.org/sdn-resources/technical-library>
- Versión 1.5.1 Abril de 2015
- Probablemente OF 1.0 sea lo más implementado en hardware
- Las siguientes versiones han ido introduciendo mejoras, más flexibilidad, pero también haciéndolo más complejo
- OF 1.1
- OF 1.2
  - Soporte de campos de IPv6, ICMPv6, ND
  - Mejora la extensibilidad de las reglas de *match*
- (...)



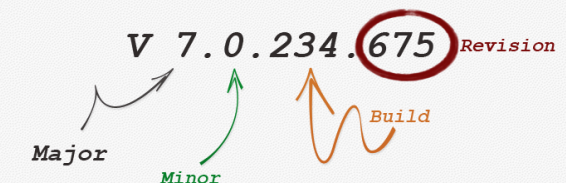
# Versiones

- <https://www.opennetworking.org/sdn-resources/technical-library>
- Versión 1.5.1 Abril de 2015
- Probablemente OF 1.0 sea lo más implementado en hardware
- Las siguientes versiones han ido introduciendo mejoras, más flexibilidad, pero también haciéndolo más complejo
- OF 1.1
- OF 1.2
- OF 1.3.x
  - *Meters* por flujo (limitadores para QoS)
  - Soporte de PBB
- (...)



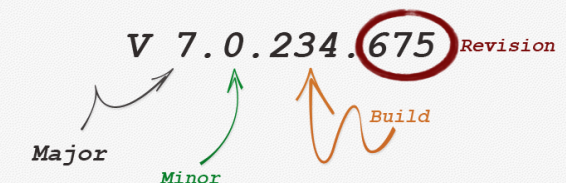
# Versiones

- <https://www.opennetworking.org/sdn-resources/technical-library>
- Versión 1.5.1 Abril de 2015
- Probablemente OF 1.0 sea lo más implementado en hardware
- Las siguientes versiones han ido introduciendo mejoras, más flexibilidad, pero también haciéndolo más complejo
- OF 1.1
- OF 1.2
- OF 1.3.x
- OF 1.4
  - Mayor extensibilidad
  - Soporte de puertos ópticos (frecuencias, potencia, etc)
- (...)



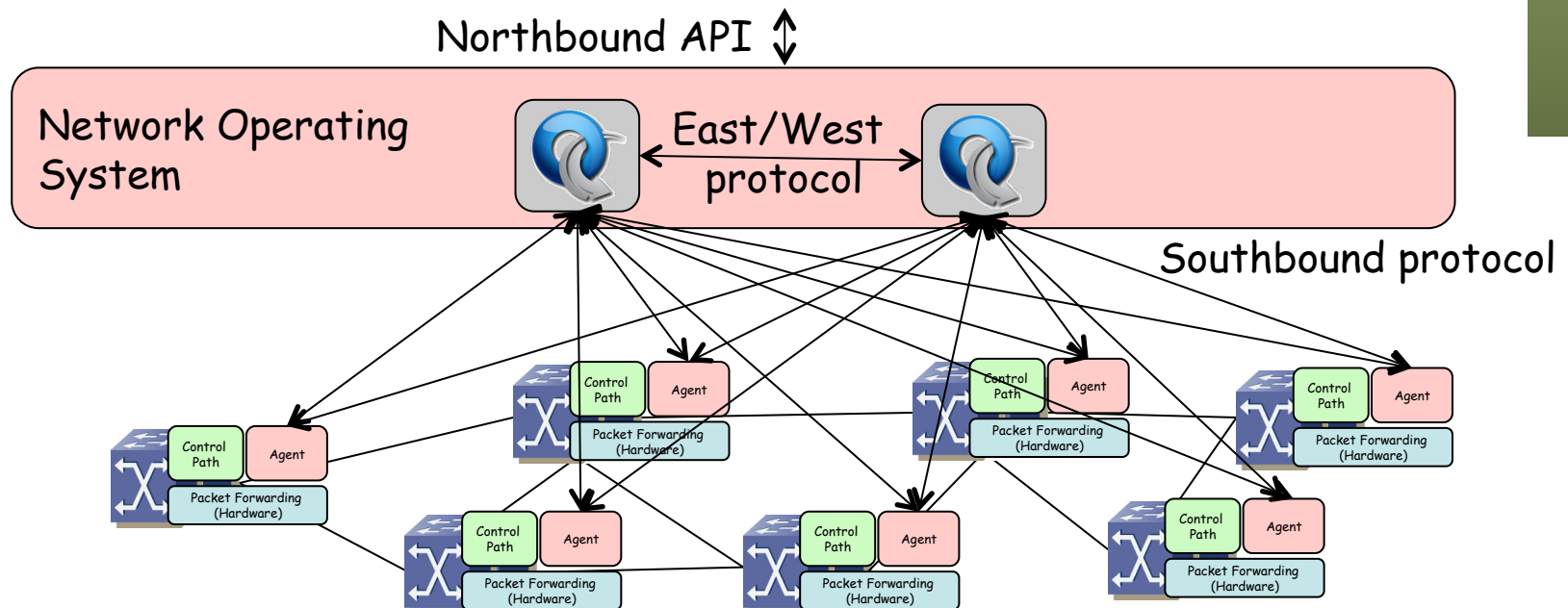
# Versiones

- <https://www.opennetworking.org/sdn-resources/technical-library>
- Versión 1.5.1 Abril de 2015
- Probablemente OF 1.0 sea lo más implementado en hardware
- Las siguientes versiones han ido introduciendo mejoras, más flexibilidad, pero también haciéndolo más complejo
- OF 1.1
- OF 1.2
- OF 1.3.x
- OF 1.4
- OF 1.5
  - *Egress tables*
  - Soporte para más que Ethernet
  - Flags TCP



# APIs

- OpenFlow es un *Southbound API*
- El ONF asocia OpenFlow a SDN pero una SDN no necesita emplear necesariamente OpenFlow
- Podríamos considerar OF a día de hoy el API south estándar
- No hay *Northbound API* estandarizada, ni *de facto*
- No hay *East/West API* estandarizada



# Controladores

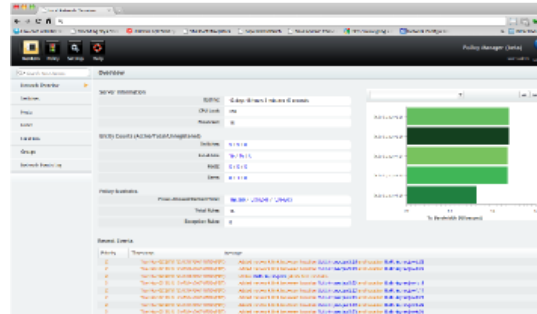


- NOX
  - <http://www.noxrepo.org>
  - Desarrollado por Nicira, cedido el código en 2008
  - Ofrece un API C++ para OF 1.0
  - Muchos otros heredan de su código
  - Incluye componentes de ejemplo para descubrir la topología, implementar un puente transparente y un switch distribuido
  - Open Source
- POX
  - Hereda de NOX
  - Permite el desarrollo en Python
  - Open Source
- Beacon
  - <https://openflow.stanford.edu/display/Beacon/Home>
  - Java (desarrollo con eclipse)
  - Open Source

# Controladores

- SNAC

- <http://www.openflowhub.org/display/Snac/SNAC+Home>
- Incluye GUI web
- Incluye un lenguaje de definición de políticas
- Open Source

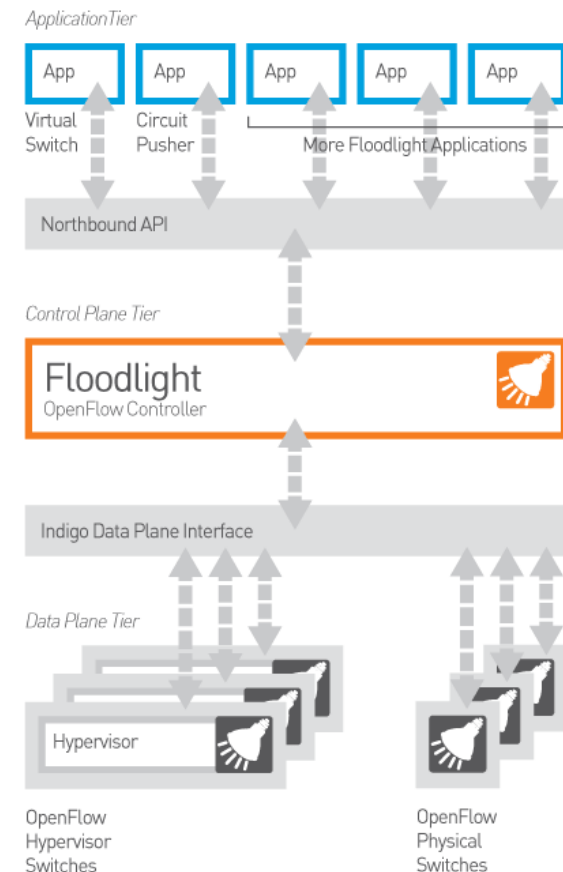


- FloodLight

- <http://www.projectfloodlight.org/floodlight/>
- Basado en Java (basado en Beacon)
- Apoyado por Big Switch Networks
- Lo emplean para construir su controlador
- Open Source



Guido Appenzeller



# VMware

- Controlador propietario
- vCenter Server controla los VDS (Virtual Distributed Switches)
- Otros componentes: vSphere, vCloud Director, vCloud Networking and Security, vCloud Automation Center, vCenter Site Recovery Manager, vCenter Operations Management Suite, vFabric Application Director for Provisioning
- Máximos vSphere 6.0:
  - 1024 VMs por host
  - 10 vNICs por VM
  - 1000 hosts por VDS
  - 1016 puertos de VDS activos por host
  - 60.000 puertos por VDS
  - 1000 hosts, 10.000 VMs en funcionamiento y 128 VDS por vCenter
  - 65.536 direcciones MAC por vCenter
  - 4/8 operaciones vMotion simultáneas por host por NIC 1/10Gbps
  - 16 VDS por host
  - etc





# Nicira

- Fundada en 2007
- Miembro fundador del ONF
- En 2011 empieza a distribuir su NVP (*Network Virtualization Platform*)
- Es un controlador para OVS (Open vSwitch)
- No emplea solo OF sino OVSDb (Open vSwitch DataBase Management Protocol)
- Adquirida en 2013 por VMware (por unos 1260 millones de \$)



Martin Casado



# Otro software

- Frameworks
  - Onix, Trema, Maestro, Ryu
  - Indigo (para añadir OF a switches)
- FlowVisor:
  - <https://github.com/OPENNETWORKINGLAB/flowvisor/wiki>
  - Actúa como un proxy entre los switches y los controladores OF
  - Permite repartir recursos de la red entre varios controladores
- ONOS
  - <http://onosproject.org/>
  - Open Network Operating System
- Avior, Oflops, Cbench, Twister, FortNOX, LINC, Pantou, Of13softswitch, Cisco OnePK, Plexxi, etc etc etc
- ¡Se abrió la veda al software!



NFV

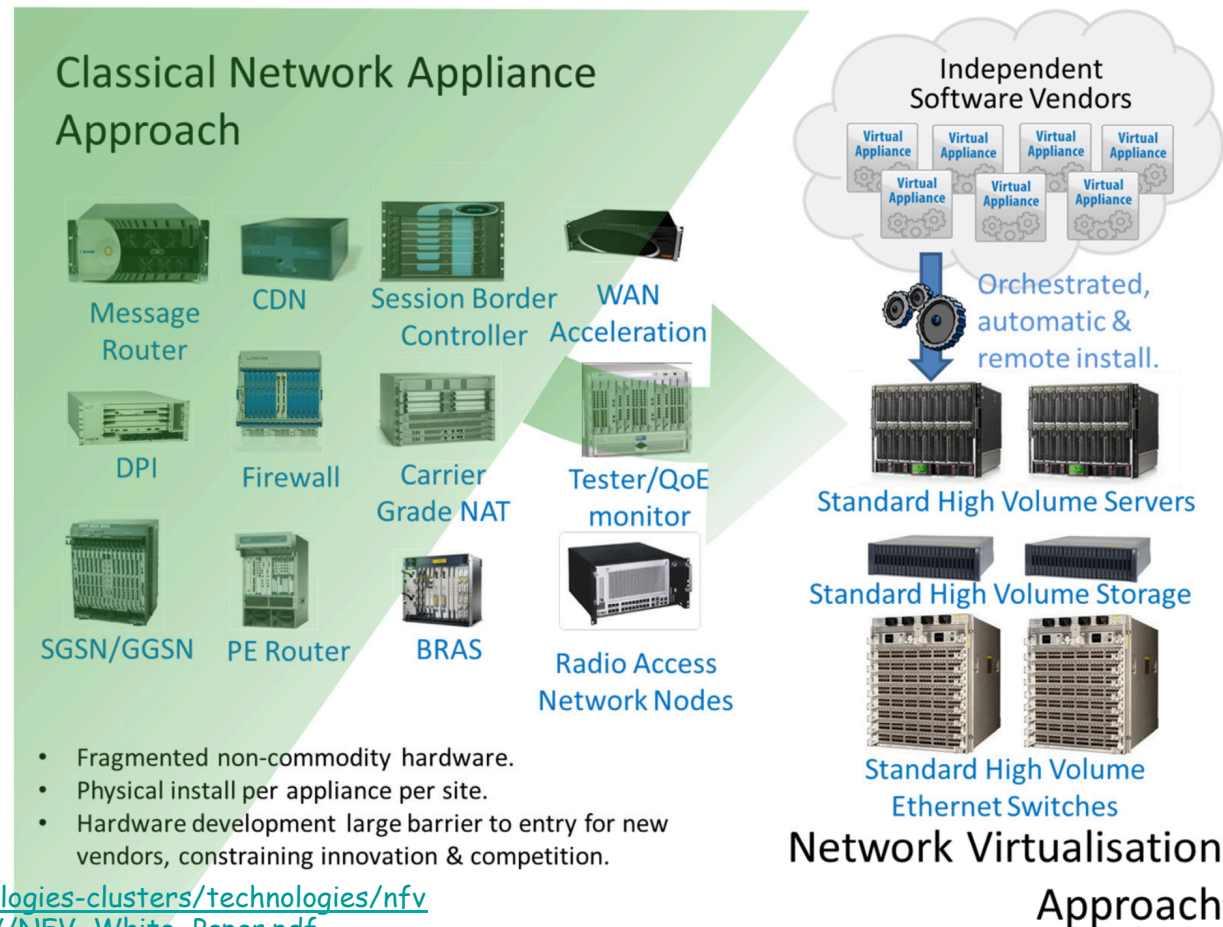
# El problema

- Problema de las operadoras
- Gran cantidad de *appliances*
- Desplegar un nuevo servicio requiere espacio y alimentación para ese nuevo hardware
- Nuevas habilidades de la gente para diseñar, integrar y operar el servicio con ese nuevo hardware
- Ese hardware alcanza su límite de vida con rapidez, lo cual requiere políticas de remplazo que no crean nuevo beneficio
- Los operadores declaran no estar incrementando sus beneficios pero aumentan sus costes (más tráfico, más servicios)



# NFV

- *Network Functions Virtualisation* (complementario a SDN)
- Se busca mover de hardware dedicado a máquinas virtuales
- Un ISG (*Industry Specification Group*) de ETSI desde finales de 2012
- Hoy más de 200 compañías

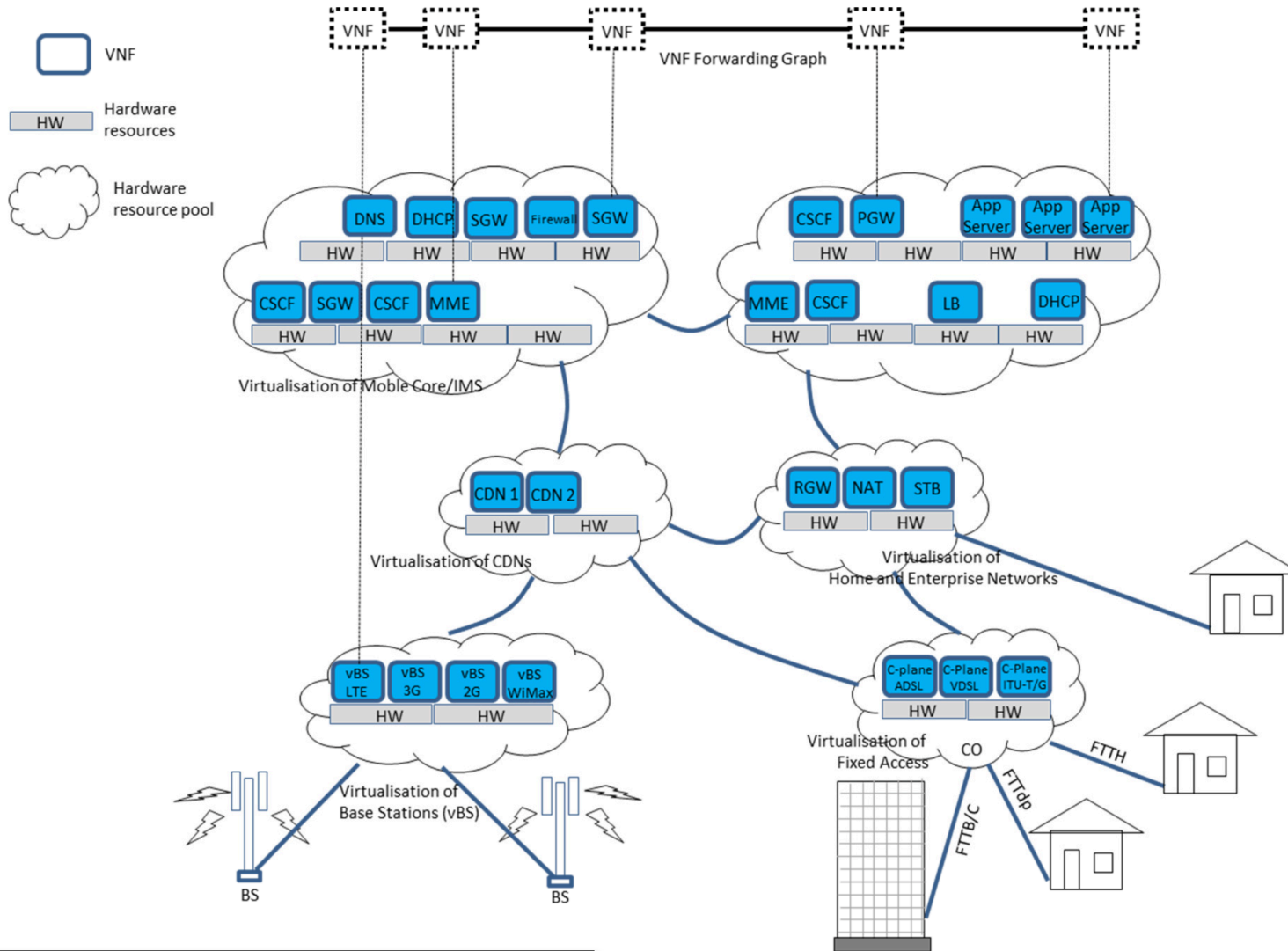


<http://www.etsi.org/technologies-clusters/technologies/nfv>  
[https://portal.etsi.org/NFV/NFV\\_White\\_Paper.pdf](https://portal.etsi.org/NFV/NFV_White_Paper.pdf)

# Use cases

- Switching elements: BNG, CG-NAT, routers.
- Mobile network nodes: HLR/HSS, MME, SGSN, GGSN/PDN-GW, RNC, Node B, eNode B.
- Functions contained in home routers and set top boxes to create virtualised home environments.
- Tunnelling gateway elements: IPSec/SSL VPN gateways.
- Traffic analysis: DPI, QoE measurement.
- Service Assurance, SLA monitoring, Test and Diagnostics.
- NGN signalling: SBCs, IMS.
- Converged and network-wide functions: AAA servers, policy control and charging platforms.
- Application-level optimisation: CDNs, Cache Servers, Load Balancers, Application Accelerators.
- Security functions: Firewalls, virus scanners, intrusion detection systems, spam protection.

# Ejemplos



VNF = Virtualised Network Function

# Algunos beneficios

- Reducción de coste de equipos
- Reducción de consumo eléctrico
- Reducción de tiempo de despliegue de un nuevo servicio
- Posibilidad de tener servicios en producción, prueba y desarrollo en la misma infraestructura
- Escalado rápido del servicio
- Abre el mercado a desarrolladores de soft (no necesitan desarrollar hardware)
- Multi-tenancy
- Mejores habilidades existentes para la gestión de infraestructura IT de gran escala que de equipos de red
- Reducción de tiempos de reparación
- Reducción de tiempos de actualización de software
- Etc etc





# Facilitadores

- *Cloud Computing*
  - Virtualización (hypervisores, vSwitch, smart NICs)
  - *Orchestration*
  - Open APIs
- Grandes volúmenes de servidores
  - Componentes estándar (por ejemplo x86), vendidos por millones (escala) e intercambiables (competencia)
  - En lugar de *appliances* que dependen de ASICs



# Ejemplo: B4N CG-NAT

## B4N CG-NAT SPECIFICATIONS

### WHAT IS B4N CG-NAT?

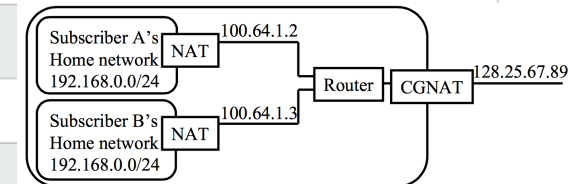
B4N CG-NAT is an NFV-based solution designed to provide high performance and transparent address and protocol translation. B4N CG-NAT helps Service Providers to mitigate IPv4 address exhaustion by using address and port translation in large scale and provides native integration within existent operators' infrastructure.

B4N CG-NAT is a fully virtualized and SDN ready solution that utilizes commodity x86 servers and provides carrier grade performance by using Intel® Data Plane Development Kit libraries.

CG-NAT solution provides maximum **500Gbps** throughput performance and fully compliant with **ETSI NFV ISG architecture**

	CONF.10	CONF.50	CONF.500	DISTRIBUTED
	10 Gbps	50 Gbps	500 Gbps	Unlimited
Max Throughput				
Connections per Second	200K per 10Gbps			Depends on OpenFlow switches performance, but not less than BOXED
Two-way sessions	10M per 10Gbps			
Resiliency	N+1. Active-Active, Active-Standby			
Supported protocols	NAT44 PCP			
Interfaces	REST API NETCONF			
Management	WEB CLI			
Supported hypervisors	LXC (Linux Containers) KVM VMware			
Logging	Local or external SYSLOG Server			

The diagram illustrates a network architecture for two subscribers. Subscriber A's Home network (192.168.0.0/24) and Subscriber B's Home network (192.168.0.0/24) are connected via NAT to a Router. The Router then connects to a CGNAT block, which finally connects to the Internet (128.25.67.89). The IP addresses 100.64.1.2 and 100.64.1.3 are shown as intermediate addresses between the NAT blocks and the Router.



#### SCALABILITY

Simple extend capacity and performance by adding new B4N NAT VNFs and Distributed Switches, while maintaining existent network architecture



#### COMMODITY HARDWARE

Using commodity x86 servers instead of dedicated hardware devices



#### UNIFIED MANAGEMENT

Single point of management through powerful WEB-interface



#### AUTOMATION

B4N CG-NAT provides set of tools for automate service management



#### CONFORMANCE WITH REFERENCE ARCHITECTURE

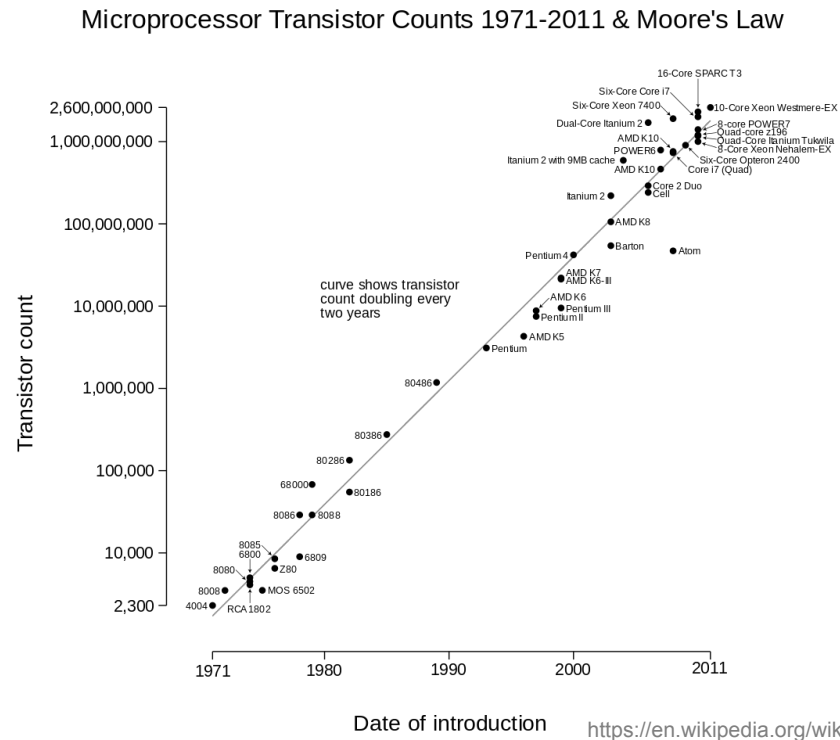
Fully compliant with MANO Framework. B4N CG-NAT includes VNF-manager that can be integrated with Customer orchestration and management system.



#### SDN READY

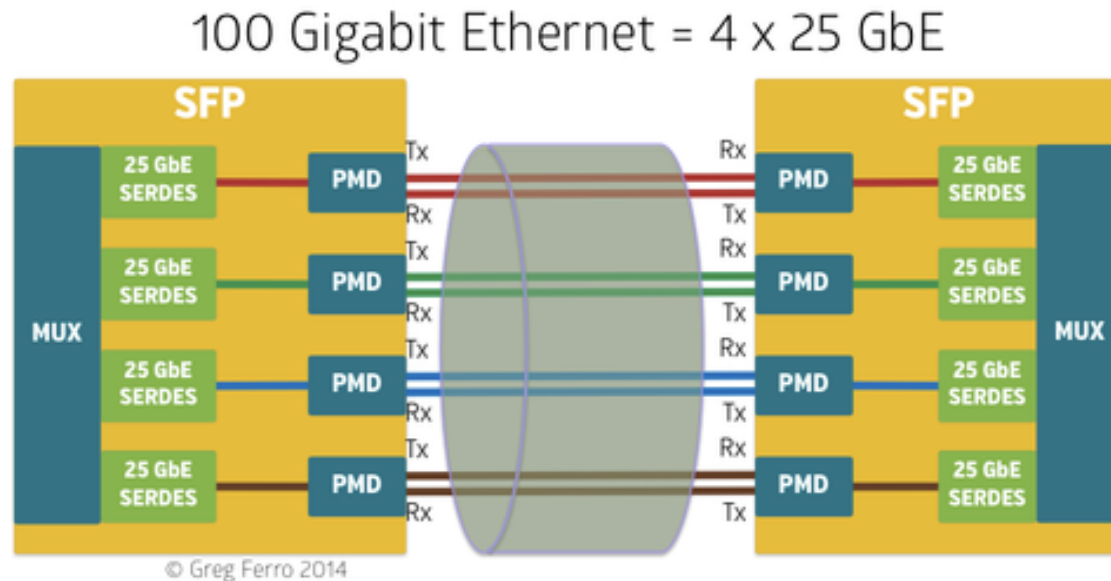
B4N CG-NAT designed to be easy integrated with Customer SDN infrastructure

# Networking hardware y el software



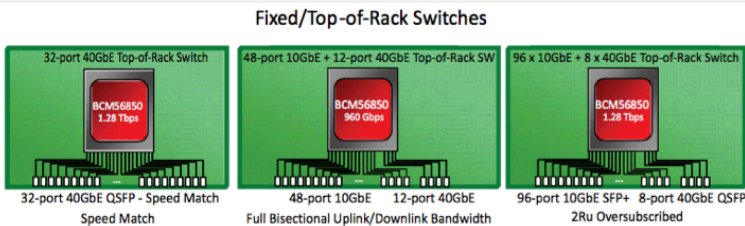
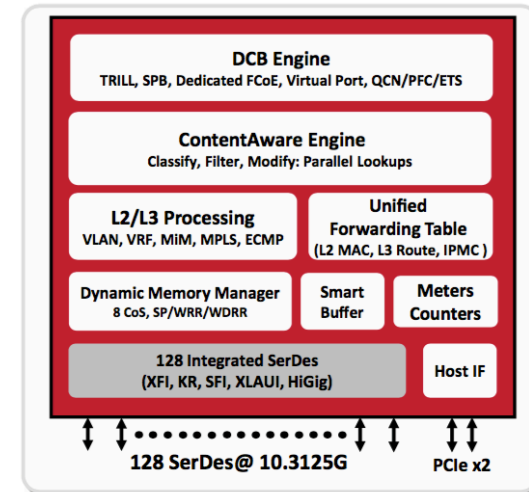
# Evolución

- Los fabricantes de equipos de red están adoptando los ritmos de producción de electrónica
- Empujados por pocos grandes clientes
- Por ejemplo: donde teníamos SerDes a 10Gbps los tendremos este año a 25Gbps, al mismo coste
- Esto permite interfaces 100GE donde antes teníamos 40GE, al mismo precio
- A día de hoy SoC (Switch on Chip) a 3.2Tbps



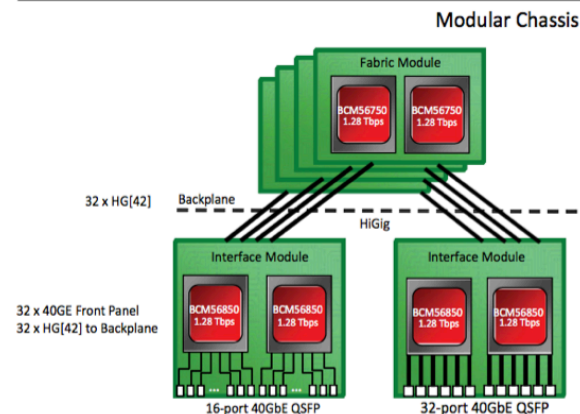
# Broadcom Trident 2

- 1.28 Tbps con puertos 10GE/40GE
- 128 SerDes 10GE (así que un máximo de 32 puertos 40GE en base a 4x10GE)
- Cut-through y Store&Forward
- VXLAN, NVGRE, 802.1Qbg EVR, 802.1BR
- Per VM traffic shaping
- DCB PFC, QCN y ETS. FCoE
- MPLS, VPLS, ISATAP, MAC-in-MAC, TRILL, SPB, Q-in-Q



## Solution Characteristics

- Single-chip design
- Lowest power/highest density
- Line rate and oversubscribed configurations for power/performance tradeoff
- FCoE support on all ports
- No external PHY needed for 10GbE or 40GbE



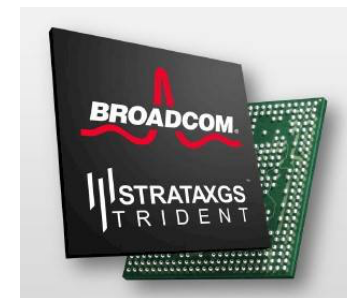
## Solution Characteristics

- Line rate and oversubscribed configurations for power/performance tradeoff
- Lowest power/highest density 40GbE solution available
- FCoE support on all ports
- Support for 40GbE flows

<https://www.broadcom.com/collateral/pb/56850-PB03-R.pdf>

# Broadcom Tomahawk

- Conmutación a 3.2 Tbps para paquetes a partir de 250 bytes
- Para paquetes de 64 bytes da un throughput de 2 Tbps
- 32 x 100GE, cada uno divisible en 4x10GE, 4x25GE, 2x50GE o 1x40GE
- SerDes 25Gbps
- 10 colas por puerto
- Bridging de VXLAN a VLAN (no routing)
- NVGRE, MPLS, SPB
- Latencia de 300-500 ns



# Trident 2 y Tomahawk

- Memoria (SRAM, TCAM) particionable para diferentes usos del switch (muchas MACs, muchas rutas IPv4, etc)



**Table 1.** Broadcom Trident 2 Forwarding Tables

Mode	Dedicated Layer 2	Shared Memory bank 1	Shared Memory bank 2	Shared Memory bank 3	Shared Memory bank 4	Host Route Dedicated	LPM Dedicated
Mode 0	32,000	Layer 2 (64,000)	Layer 2 (64,000)	Layer 2 (64,000)	Layer 2 (64,000)	Layer 3 (16,000)	16,000
Mode 1	32,000	Layer 2 (64,000)	Layer 2 (64,000)	Layer 2 (64,000)	Layer 3 (40,000)	Layer 3 (16,000)	16,000
Mode 2	32,000	Layer 2 (64,000)	Layer 2 (64,000)	Layer 3 (32,000)	Layer 3 (40,000)	Layer 3 (16,000)	16,000
Mode 3	32,000	Layer 2 (64,000)	Layer 3 (32,000)	Layer 3 (32,000)	Layer 3 (40,000)	Layer 3 (16,000)	16,000
Mode 4	32,000	LPM (32,000)	LPM (32,000)	LPM (32,000)	LPM (32,000)	Layer 3 (16,000)	16,000

**Table 2.** Broadcom Tomahawk Forwarding Tables

Mode	Dedicated Layer 2	Shared Memory bank 1	Shared Memory bank 2	Shared Memory bank 3	Shared Memory bank 4	Host Route Dedicated	LPM Dedicated
Mode 0	8000	Layer 2 (32,000)	Layer 2 (32,000)	Layer 2 (32,000)	Layer 2 (32,000)	Layer 3 (8000)	16,000
Mode 1	8000	Layer 2 (32,000)	Layer 2 (32,000)	Layer 2 (32,000)	Layer 3 (32,000)	Layer 3 (8000)	16,000
Mode 2	8000	Layer 2 (32,000)	Layer 2 (32,000)	Layer 3 (32,000)	Layer 3 (32,000)	Layer 3 (8000)	16,000
Mode 3	8000	Layer 2 (32,000)	Layer 3 (32,000)	Layer 3 (32,000)	Layer 3 (32,000)	Layer 3 (8000)	16,000
Mode 4	8000	LPM (32,000)	LPM (32,000)	LPM (32,000)	LPM (32,000)	Layer 3 (8000)	16,000



# Cisco ASE-2

- ACI Spine Engine 2 (ACI = Aplicacion Centric Infrastructure)
- 3.6Tbps (todos los tamaños de paquetes)
- 36x100GE, 72x40GE, 144x25GE
- 16K VRF, 32 SPAN, 64 mcast, 4K NAT
- Push/swap 5 etiquetas VPN
- DWRR con 16 colas por puerto
- WRED, ACN, AFD (Approximate Fair Dropping)

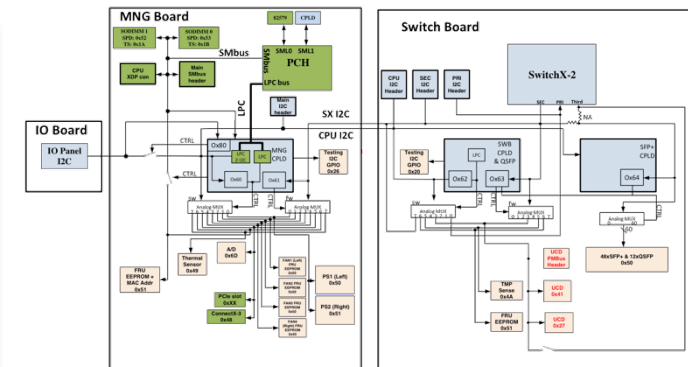
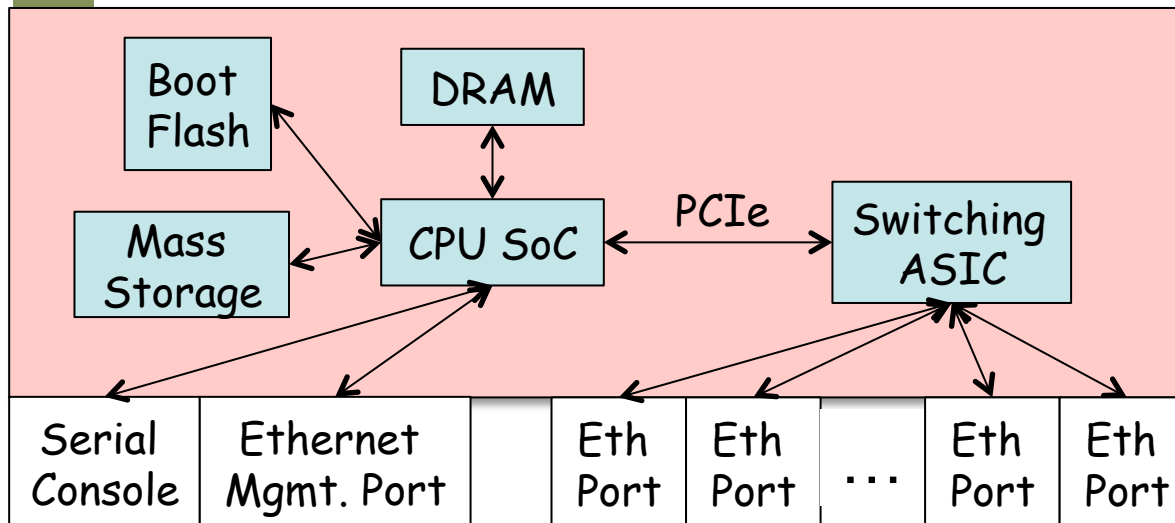
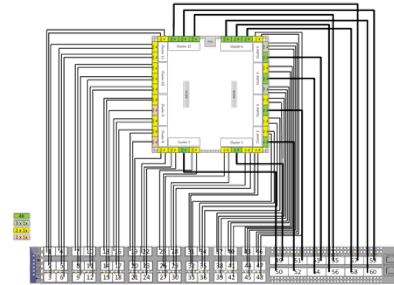
# ¿Evolución?

- La infraestructura se está simplificando
- Principalmente el hardware, controlable por software
- *White boxes* no solo servidores sino también switches
- También se venden ya switches “*Bare metal*” = solo el hardware



# Bare metal switches

- Menores costes
- El mismo equipo un día es un switch, otro un firewall, otro un balanceador... dentro de las limitaciones del ASIC
- Ejemplo:
  - Open Compute Networking Project
  - <http://www.opencompute.org/wiki/Networking>
  - Especificaciones completas de conmutadores
- Fabricantes: Mellanox, Quanta, Penguin Computing, Edge-core, Acton, Dell, etc



# ¿Evolución?

- Para estos equipos sistemas operativos y gran cantidad de software, generalmente basados en linux, muchos de código abierto
- Ejemplos:
  - Open Network Install Environment (ONIE): <http://onie.opencompute.org>
  - Open Network Linux: <http://opennetlinux.org>
  - Big Switch's Switch Light OS
  - Pica8 PicOS
  - Cumulus Linux
- Es decir, igual que en el entorno de servidor, puedes cambiar el hardware, instalar el sistema operativo que quieras y desarrollar tus aplicaciones (...)



# ¿Evolución?

- Para diferenciarse, los proveedores desarrollan software propietario para ofrecer sus servicios
- Porque hoy en día ya es el software por lo que principalmente están cobrando los fabricantes “no-open”
- Muchos modelos ToR de fabricantes conocidos son switches bare-metal que han comprado, cambiado el software y el frontal

