upna
Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

Redes de Nueva Generación
Área de Ingeniería Telemática

OpenFlow

OpenFlow

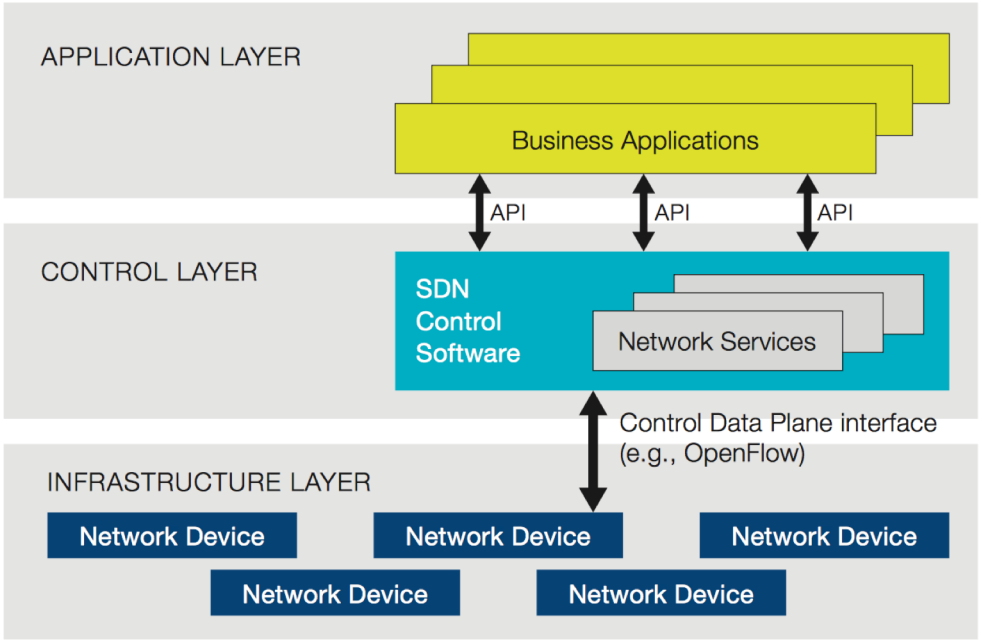
- Su origen en proyectos de investigación en la Universidad de Stanford
- En 2011 se funda el consorcio ONF
  - *Open Networking Foundation*
  - https://www.opennetworking.org
  - Más de 140 empresas (fabricantes, operadoras, ISPs, startups, etc)
- OpenFlow es un protocolo “southbound”
- No hace “nada” sin una aplicación que lo emplee

OPEN NETWORKING FOUNDATION

Martin Casado

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 though 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. "*



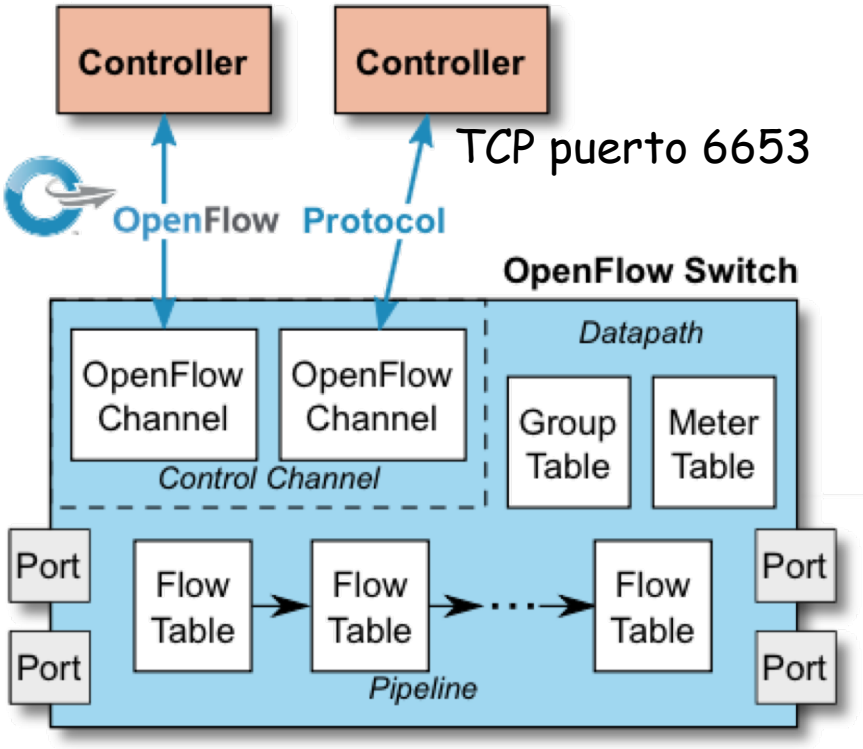
https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf

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

Flow Tables

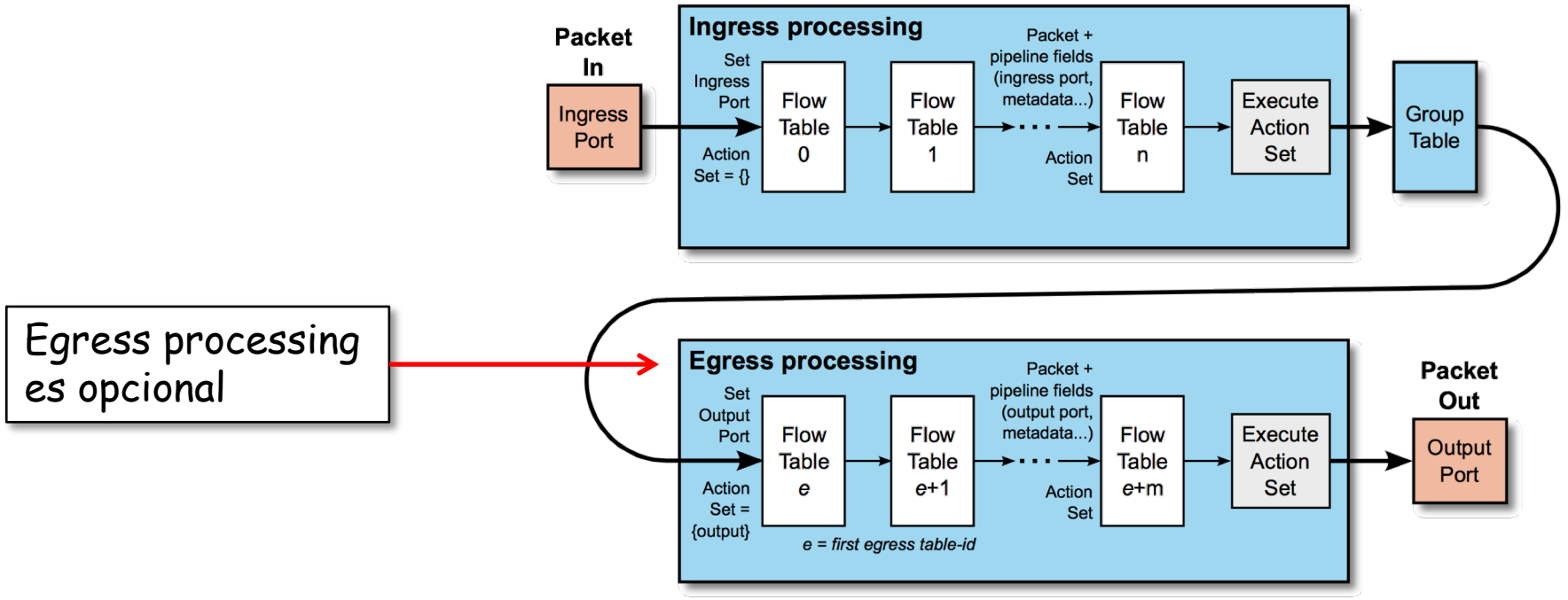
- Contienen la información sobre los campos a comprobar (*match fields*) en los paquetes y qué hacer con ellos
- El controlador puede añadir, modificar y borrar entradas empleando OF
- Las "acciones" son las operaciones en caso de que el paquete verifique la entrada en la tabla
- Puede reenviar el paquete,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



https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.0.noipr.pdf

OpenFlow pipeline

- Debe tener al menos una tabla aunque pueden ser más (desde 1.1, permite procesado de etiquetas MPLS)
- Hay procesado 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



https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.0.noipr.pdf

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	*Required*
Packet Lookups	64	*Optional*
Packet Matches	64	*Optional*
Per Flow Entry		
Received Packets	64	*Optional*
Received Bytes	64	*Optional*
Duration (seconds)	32	*Required*
Duration (nanoseconds)	32	*Optional*
Per Port		
Received Packets	64	*Required*
Transmitted Packets	64	*Required*
Received Bytes	64	*Optional*
Transmitted Bytes	64	*Optional*
Receive Drops	64	*Optional*
Transmit Drops	64	*Optional*
Receive Errors	64	*Optional*
Transmit Errors	64	*Optional*
Receive Frame Alignment Errors	64	*Optional*
Receive Overrun Errors	64	*Optional*
Receive CRC Errors	64	*Optional*
Collisions	64	*Optional*
Duration (seconds)	32	*Required*
Duration (nanoseconds)	32	*Optional*
Per Queue		
Transmit Packets	64	*Required*
Transmit Bytes	64	*Optional*
Transmit Overrun Errors	64	*Optional*
Duration (seconds)	32	*Required*
Duration (nanoseconds)	32	*Optional*
Per Group		
Reference Count (flow entries)	32	*Optional*
Packet Count	64	*Optional*
Byte Count	64	*Optional*
Duration (seconds)	32	*Required*
Duration (nanoseconds)	32	*Optional*
Per Group Bucket		
Packet Count	64	*Optional*
Byte Count	64	*Optional*
Per Meter		
Flow Count	32	*Optional*
Input Packet Count	64	*Optional*
Input Byte Count	64	*Optional*
Duration (seconds)	32	*Required*
Duration (nanoseconds)	32	*Optional*
Per Meter Band		
In Band Packet Count	64	*Optional*
In Band Byte Count	64	*Optional*

Entradas en las tablas

- *Instructions:*
  - Cambio al paquete, acciones, etc, cuando se selecciona la entrada
  - Las hay de implementacion 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	Cookie	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
    - 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

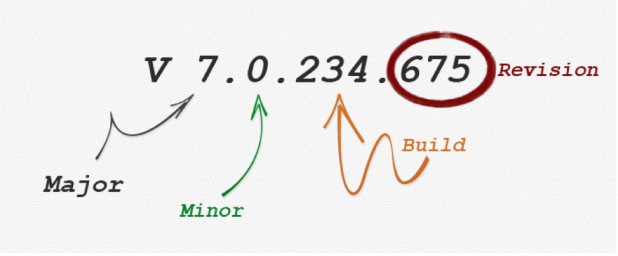
- TCP (puerto 6653), opcionalmente empleando TLS
- Hay mensajes:
  - De controlador a conmutador
  - Asíncronos (desde el conmutador)
  - Simétricos
    - Hello, al establecer la conexión
    - Echo, para comprobar que el otro extremo está vivo y tal vez para medir latencia o bw
    - Error

upna
Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa
Redes de Nueva Generación
Área de Ingeniería Telemática

OpenFlow – El protocolo y los controladores

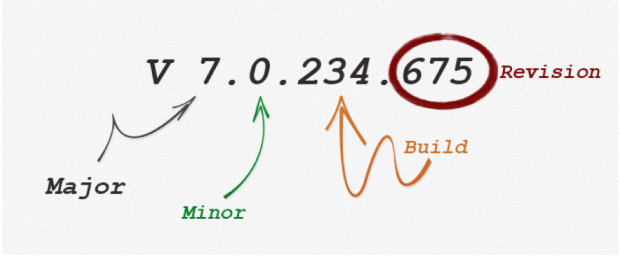
Versiones

- 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

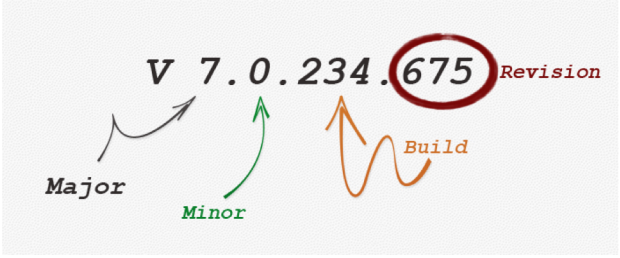
- https://www.opennetworking.org/sdn-resources/technical-library
- Versión 1.5.1 Abril de 2015
- Probablemente OF 1.0 sea lo más implementado en hardware
- Las siguientes versiones han ido introduciendo mejoras, más flexibilidad, pero también haciéndolo más complejo
- OF 1.1
- OF 1.2
- OF 1.3.x
  - *Meters*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

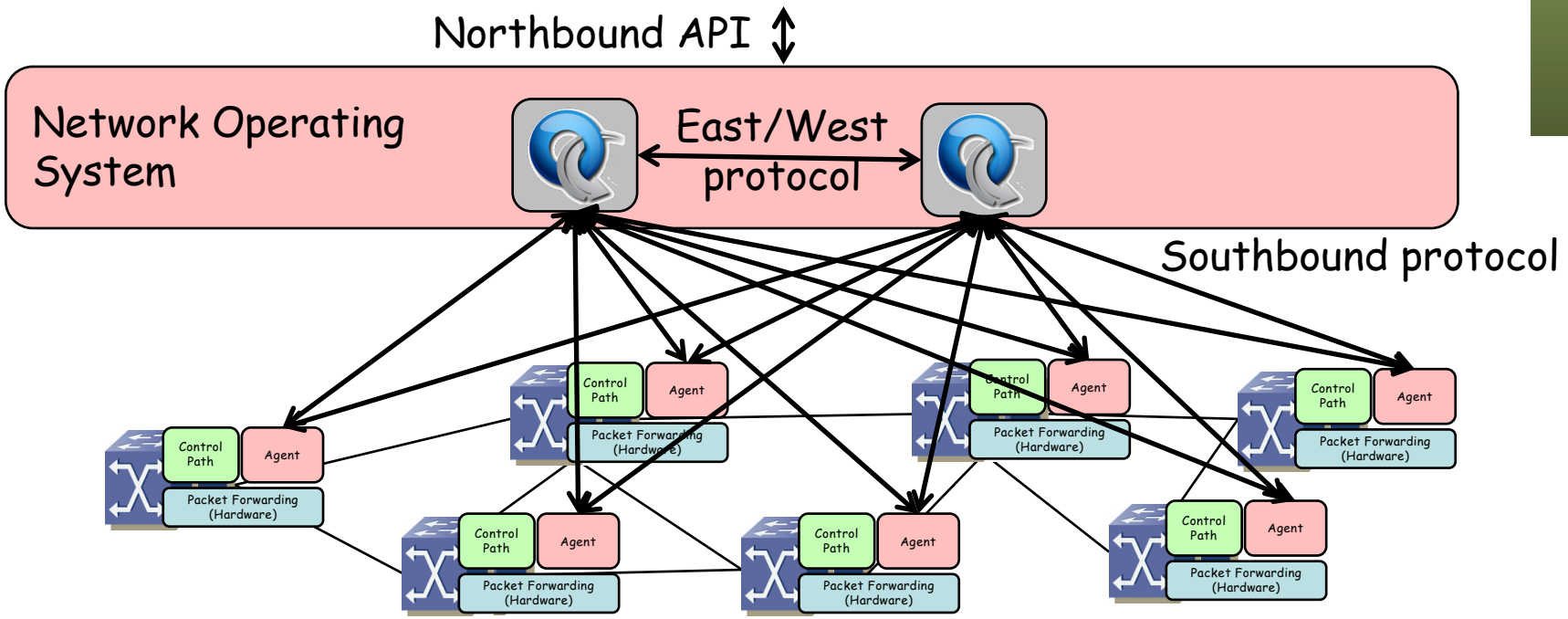
Versiones

- 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

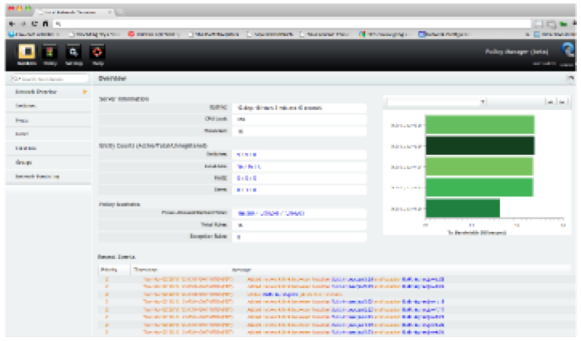


Controladores

- 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

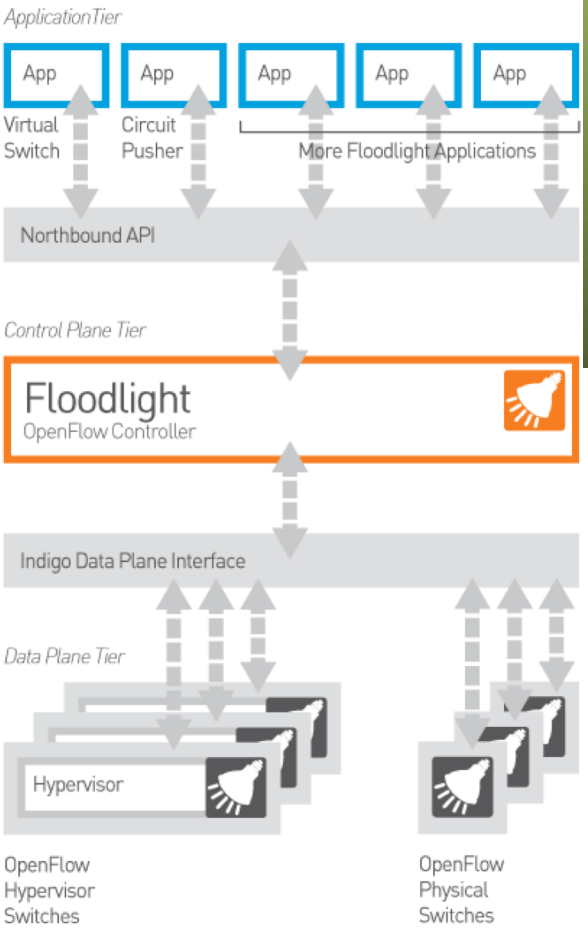
Controladores

- 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



La última década en el plano de control

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

OvS
Open vSwitch

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

http://www.vmware.com/pdf/vsphere6/r60/vsphere-60-configuration-maximums.pdf

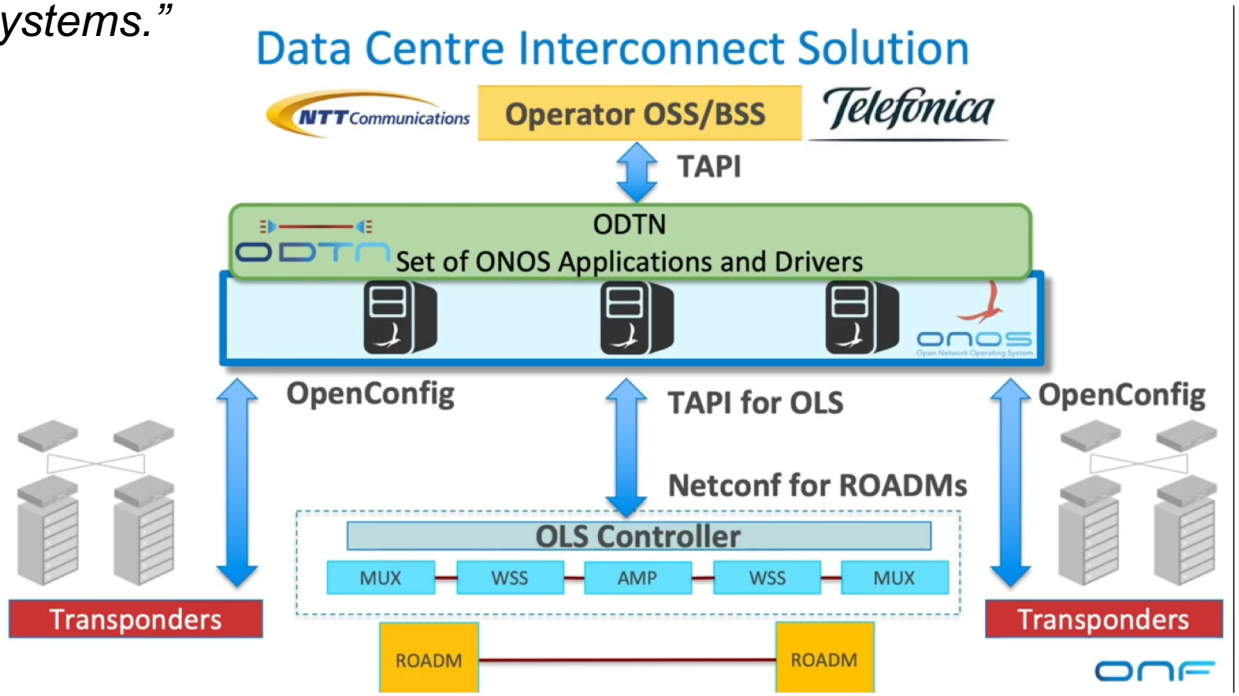
Otro software

- 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
- Avior, Oflops, Cbench, Twister, FortNOX, LINC, Pantou, Of13softwitch, Cisco OnePK, Plexxi, etc etc etc

- ¡Se abrió la veda al software!

ONOS

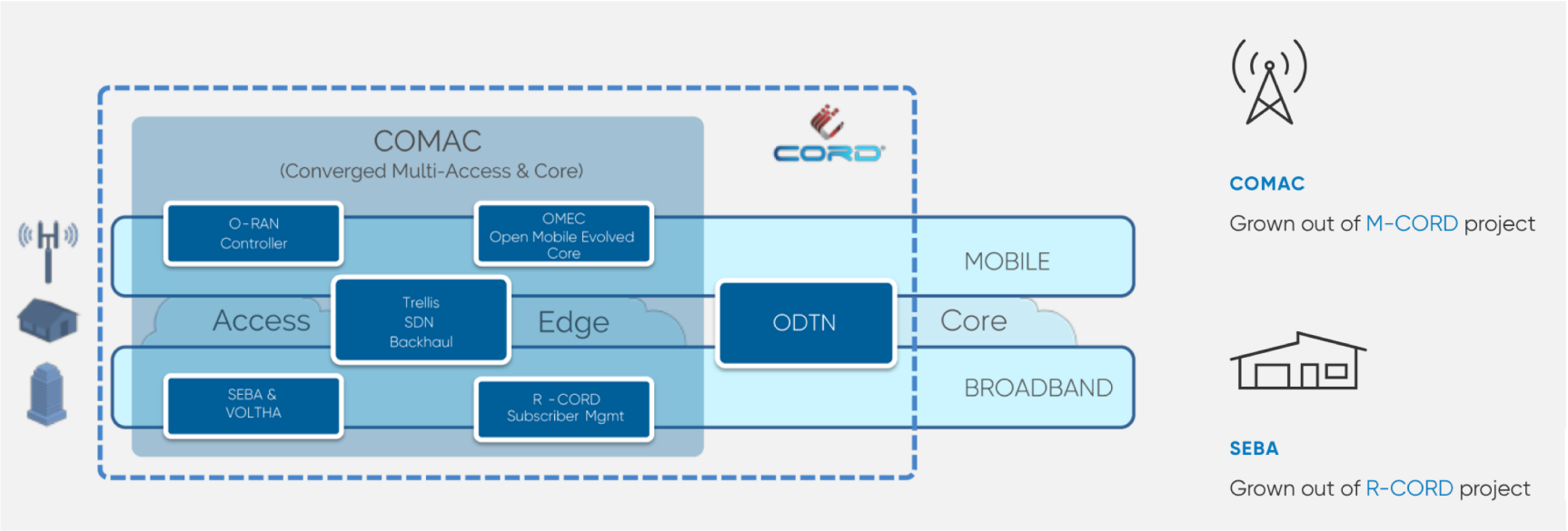
- Open Network Operating System (gestionado por el ONF) (Open Source)
- *"ONOS was designed to meet the needs of operators wishing to build carrier-grade solutions that leverage the economics of **white box merchant silicon hardware** while offering the flexibility to create and deploy **new dynamic network services** with simplified programmatic interfaces."*
- *"ONOS supports both configuration and real-time control of the network, **eliminating the need to run routing and switching control protocols inside the network fabric**."*
- *"By moving intelligence into the ONOS cloud controller, innovation is enabled and end-users can easily create new network applications without the need to alter the dataplane systems."*



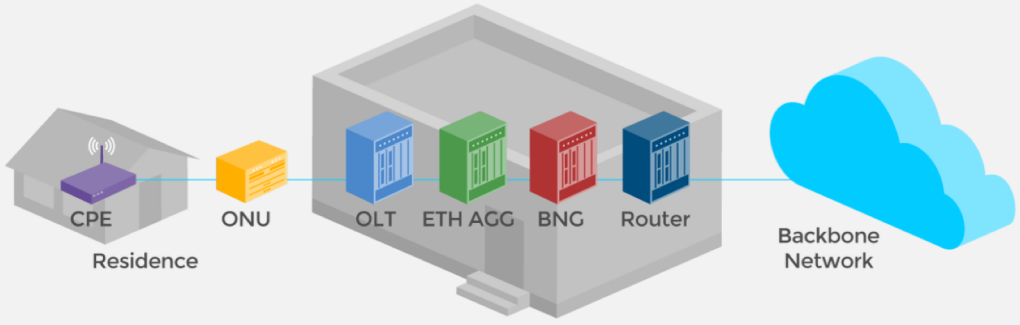
https://opennetworking.org/onos/

CORD

- Central Office Re-architected as a Datacenter
- https://opennetworking.org/cord/
- "... leverages SDN, NFV and Cloud technologies to build agile datacenters for the network edge"



R-CORD



CPE: Customer Premises Equipment
ONU: Optical Network Unit
OLT: Optical Line Termination
ETH AGG: Ethernet Aggregation Switch
BNG: Broadband Network Gateway

Legacy Central Office

Legacy networks were built using a number of discrete purpose-built hardware devices to connect residential subscribers to the carrier's backbone network. Each of these devices is a source of complexity and considerable expense, as both capex and opex. Additionally, new hardware is needed when scaling capacity or creating new services.

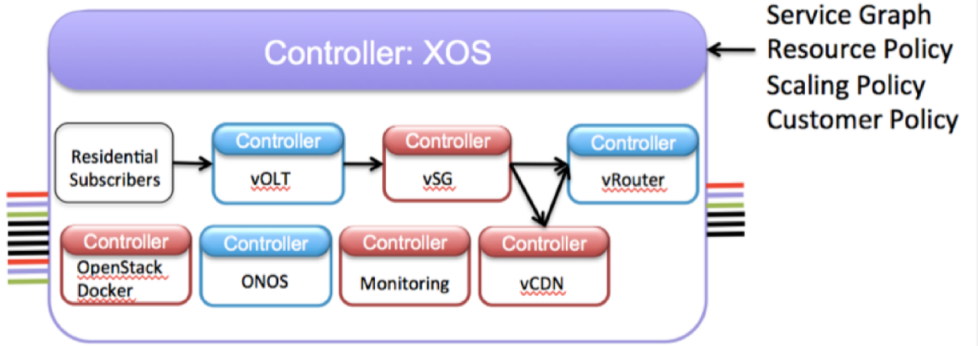
R-CORD implements the complete residential ultra-broadband solution as a cloud-native collective of virtual machines and containers running, to the greatest extent possible, on general-purpose data center infrastructure.

While specialized access hardware is needed to physically connect subscribers (via GPON, DOCSIS or similar), the VOLTHA project abstracts even this special purpose equipment to make it manageable as an OpenFlow manageable resource.

This architecture, all available as open source software, brings many of the advantages of cloud-era computing to operator access networks, including:

- CAPEX advantages of leveraging commodity hardware and dynamic workload distribution.
- OPEX advantages of cloud automation and lights-out operation.
- New revenue sources by enabling the dynamic crafting and deployment of new innovative services without hardware upgrades or truck rolls.

R-CORD Architecture

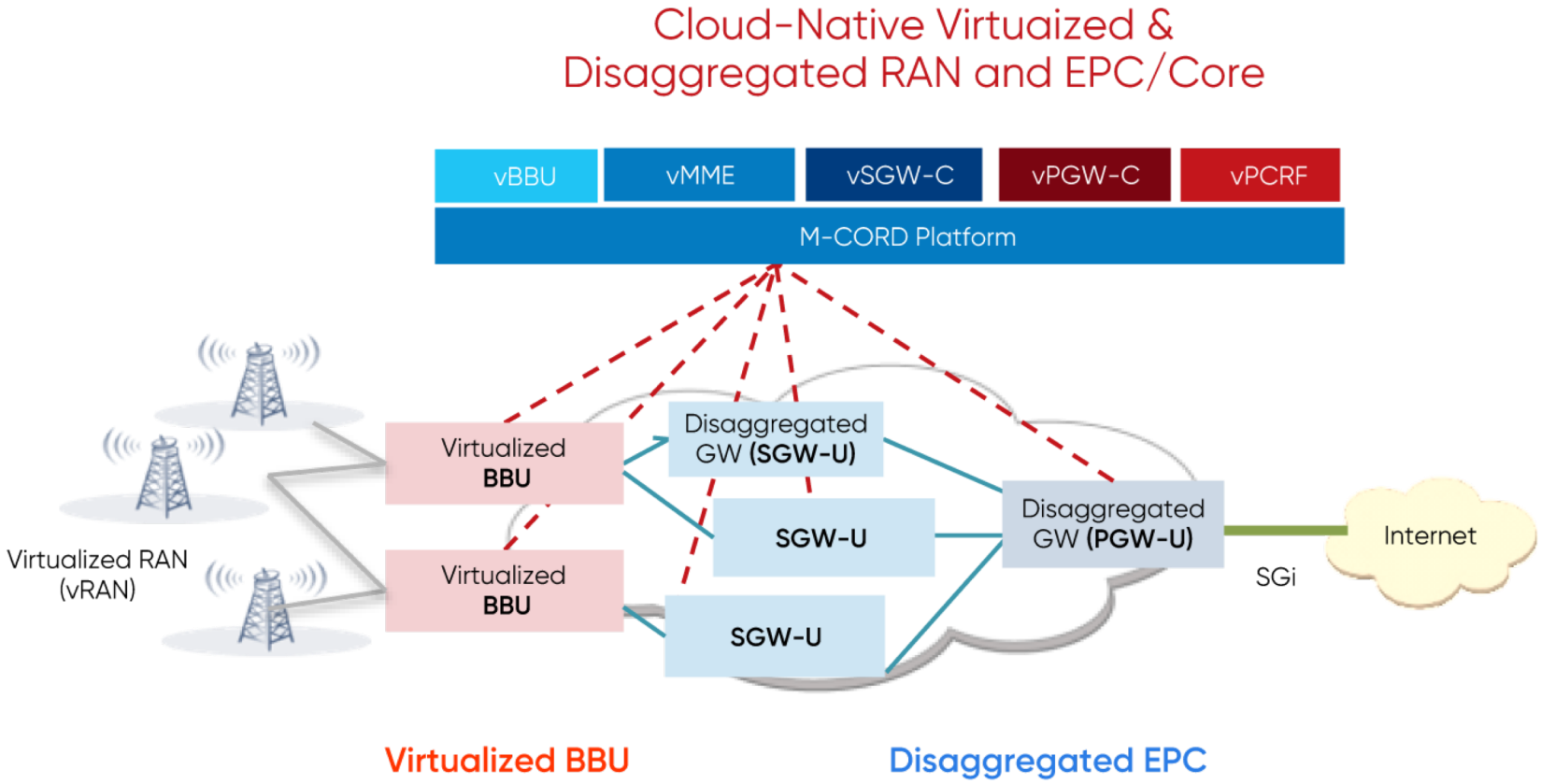


Everything-as-a-Service (XaaS) / Micro-Services Architecture

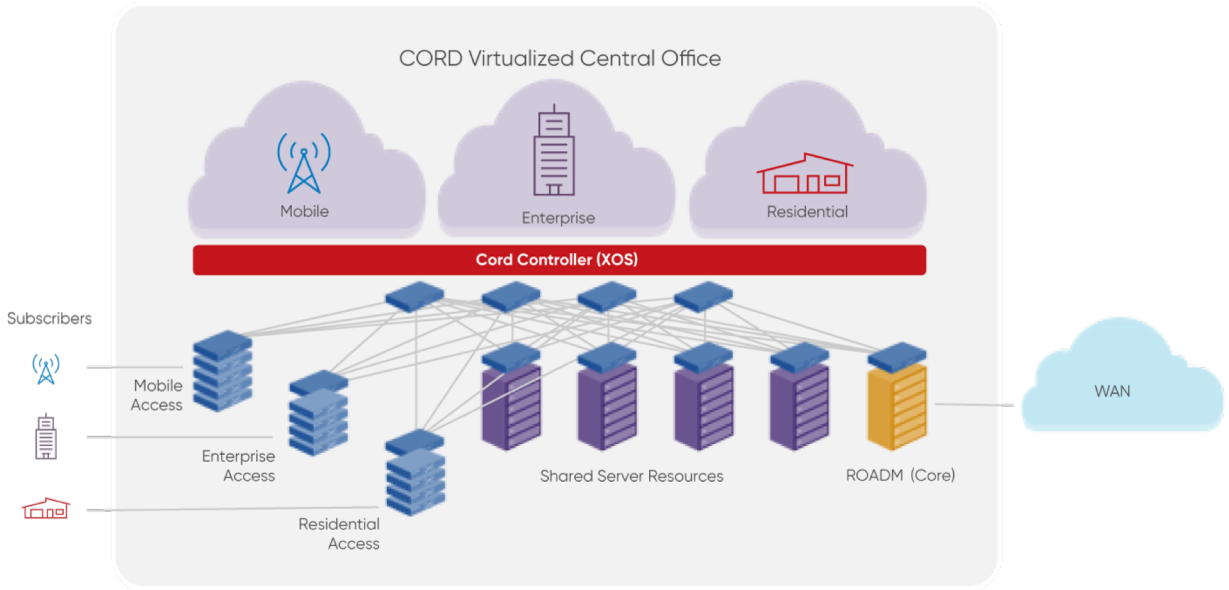
...

https://opennetworking.org/cord/

M-CORD


Cloud-Native Virtualized &
Disaggregated RAN and EPC/Core
vBBU
vMME
vSGW-C
vPGW-C
vPCRF
M-CORD Platform
Virtualized RAN
(vRAN)
Virtualized
BBU
Virtualized
BBU
Disaggregated
GW (SGW-U)
SGW-U
SGW-U
Disaggregated
GW (PGW-U)
SGi
Internet
Virtualized BBU
Disaggregated EPC

CORD



S/W Architecture

CORD includes four software subsystems:

PLATFORM:

Common base layer includes Kubernetes (container management system) and ONOS (SDN controller), with Stratum loaded on to each switch.

PROFILE:

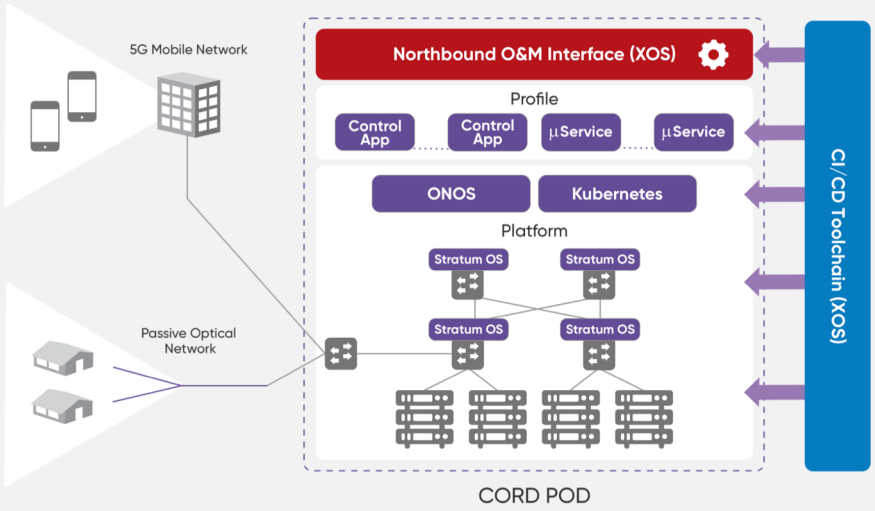
Deployment-specific collection of microservices and SDN control apps selected to run on a particular POD.

WORKFLOW:

Deployment-specific integration logic needed to operationalize the POD in a given operator's network.

CI/CD TOOLCHAIN:

Used to assemble, deploy, operate, and upgrade a particular Platform/Profile/Workflow combination.



https://opennetworking.org/cord/

CORD

- Central Office Re-architected as a Datacenter
- https://opennetworking.org/cord/
- "... leverages SDN, NFV and Cloud technologies to build agile datacenters for the network edge"

Project Affiliated Members

OPERATOR PARTNERS

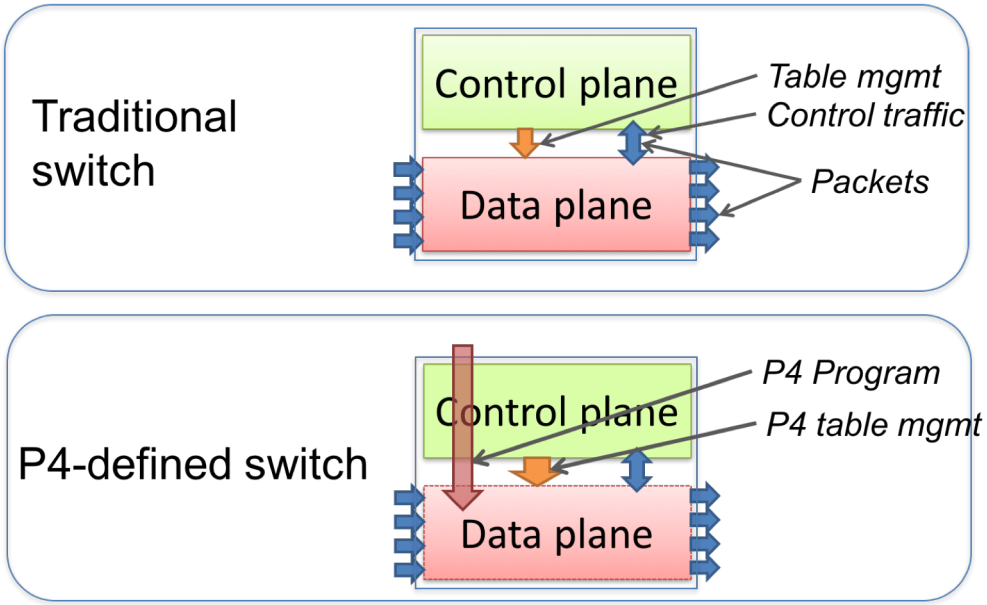
VENDOR PARTNERS

MEMBERS

Plano de datos

Planos

- Plano de control
  - Tradicionalmente atado al hardware
  - SDN lo independiza
- Plano de datos
  - En el propio diseño del hardware
  - Nuevos ASICs son reprogramables
  - Los objetos en el plano de datos no están fijos, se pueden modificar y con ellos lo que puede modificar el plano de control

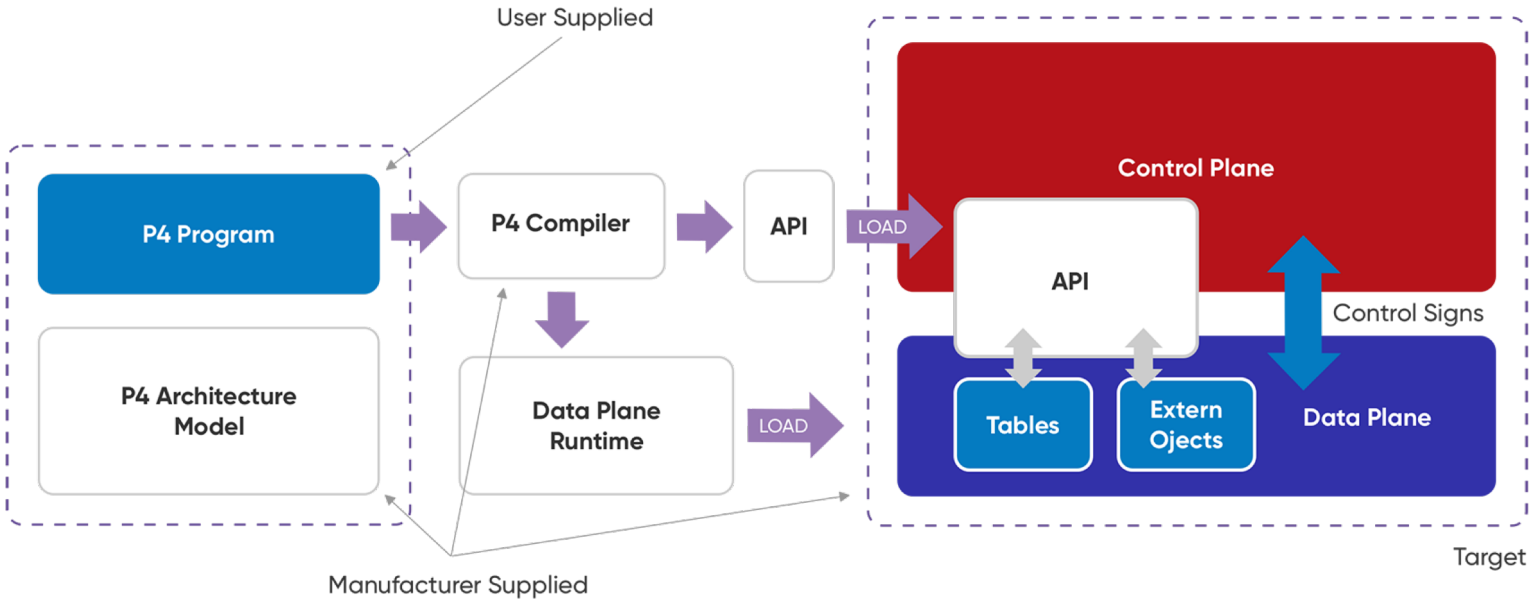


https://p4.org/p4-spec/docs/P4-16-v1.2.1.html

Ejemplo: P4

- Programming Protocol-independent Packet Processors

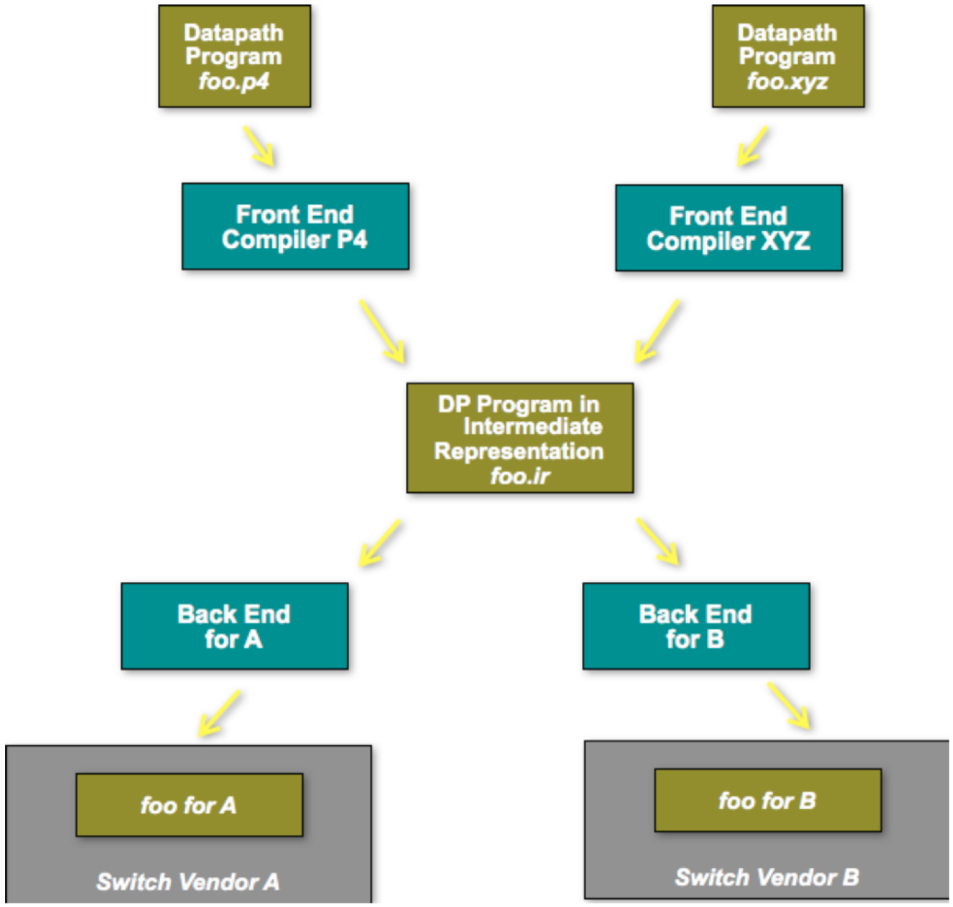
https://p4.org



https://p4.org/p4-spec/docs/P4-16-v1.2.1.html

P4

https://p4.org



https://opennetworking.org/wp-content/uploads/2013/05/TR-535_ONF_SDN_Evolution.pdf

```
parser TopParser(packet_in b, out Parsed_packet p) {
    Checksum16() ck;  // instantiate checksum unit

    state start {
        b.extract(p.ethernet);
        transition select(p.ethernet.etherType) {
            0x0800: parse_ipv4;
            // no default rule: all other packets rejected
        }
    }

    state parse_ipv4 {
        b.extract(p.ip);
        verify(p.ip.version == 4w4, error.IPv4IncorrectVersion);
        verify(p.ip.ihl == 4w5, error.IPv4OptionsNotSupported);
        ck.clear();
        ck.update(p.ip);
        // Verify that packet checksum is zero
        verify(ck.get() == 16w0, error.IPv4ChecksumError);
        transition accept;
    }
}
...
```

https://p4.org/p4-spec/docs/P4-16-v1.2.1.html