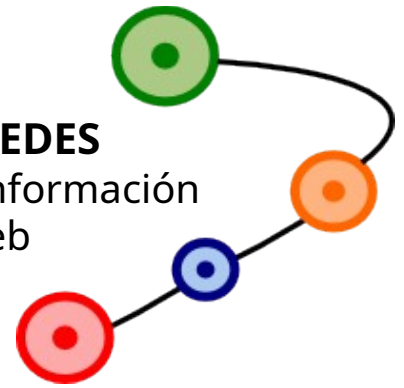




Administración y Gestión de Redes
Lic. en Sistemas de Información

Laboratorio de REDES
Recuperación de Información
y Estudios de la Web



Infraestructura de Data Centers y Cloud

Normas de Infraestructura de Data Centers
Virtualización
Software Defined Networks (SDNs)

Equipo docente:

Fernando Lorge (florge@unlu.edu.ar)

Santiago Ricci (sricci@unlu.edu.ar)

Alejandro Iglesias (aaiglesias@unlu.edu.ar)

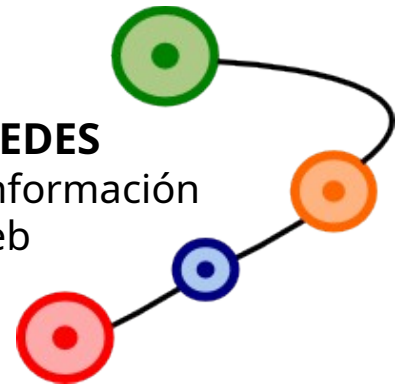
Mauro Meloni (maurom@unlu.edu.ar)

Patricio Torres (ptorres@unlu.edu.ar)



Administración y Gestión de Redes
Lic. en Sistemas de Información

Laboratorio de REDES
Recuperación de Información
y Estudios de la Web



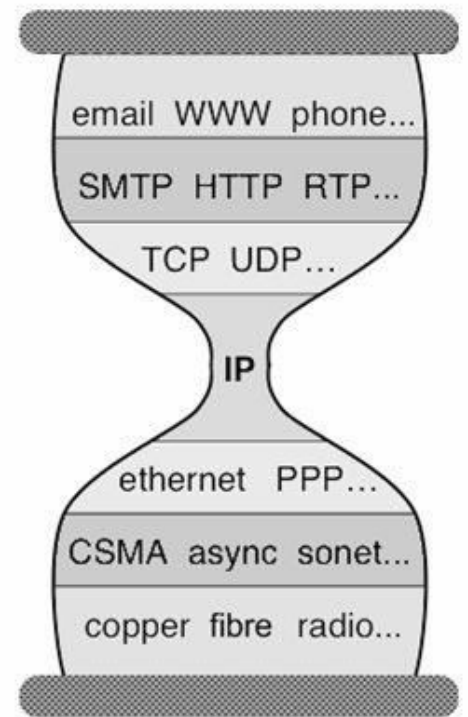
Software-Defined Networks SDN

Redes Definidas por Software
-
Openflow

Software-Defined Networking

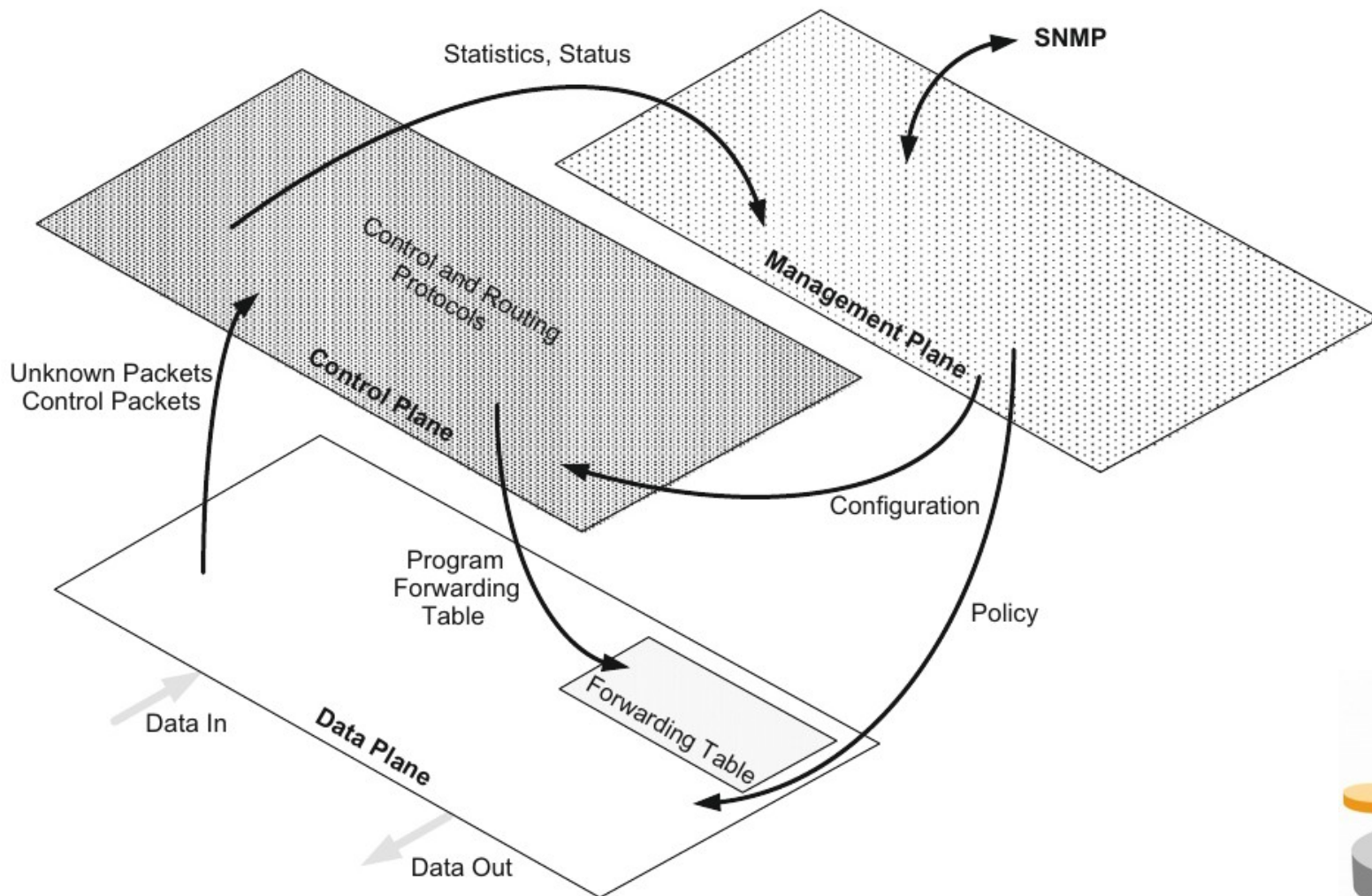
Modelo de red tradicional: El éxito de Internet

- Protocolos diseñados en capas, independientes:
 - Aplicaciones... creadas sobre...
 - Transporte fiable o no-fiable
 - Envío de paquetes "Best-effort"
 - Envío de tramas locales "Best-effort"
 - Transferencia física de bits
- Solución pensada en base a redes no-fiables para patrones de tráficos "conocidos".
- Dispositivos de red autónomos.



Software-Defined Networking

Diferentes Planos: Control, Data y Management



Software-Defined Networking

Planos tradicionales

Data Plane

- Manejo y reenvío de paquetes: switcheo, ruteo, filtrado, modificación...
- Tablas de reenvío: FIB, LFIB [1]
- Para performance se suele utilizar CAMs o TCAMs [2].
- Filters, Meters, Markers

[1] <https://mellowd.co.uk/ccie/?p=788>

[2] https://en.wikipedia.org/wiki/Content-addressable_memory



Software-Defined Networking

Planos tradicionales

Control Plane

- Configura el Data Plane indicándole como tratar los paquetes
- Acorde al hardware/software particular
- Algoritmos distribuidos (ruteo)
- Descubrimiento de topología, selección de rutas, failover...



Software-Defined Networking

Planos tradicionales

Management Plane

- Configuración del Control Plane (y posiblemente del Data Plane directamente)
 - Command Line
 - GUI
 - SNMP
 - Netconf ...



Software-Defined Networking

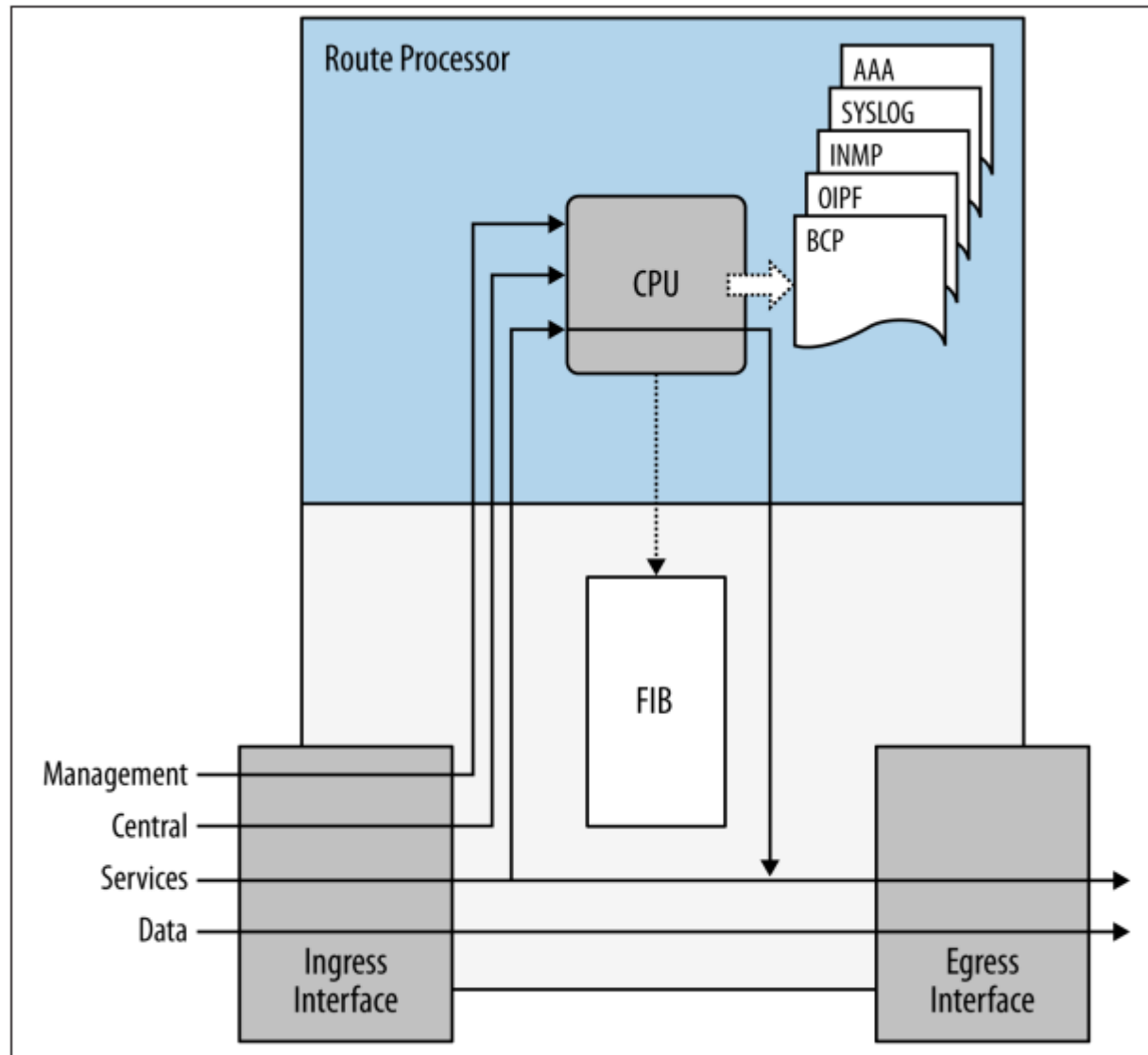


Figure 2-6. Control and data plane example implementation



Software-Defined Networking

Software-defined networks (SDN)

- Separar el Control Plane de los dispositivos y centralizarlo en un controlador...
- Administrar los servicios de red abstrayéndose del bajo nivel...
- Programar mediante una API la funcionalidad de la red...
- ¿Se usa? ¿Dónde?...



Software-Defined Networking

Software-Defined Networks (SDN):

“A set of techniques that enables to directly program, orchestrate, control and manage network resources, which facilitates the design, delivery and operation of network services in a dynamic and scalable manner.”

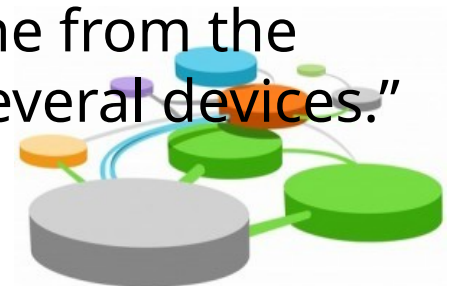
(UIT-T Y.3300)

“A programmable networks approach that supports the separation of control and forwarding planes via standardized interfaces.”

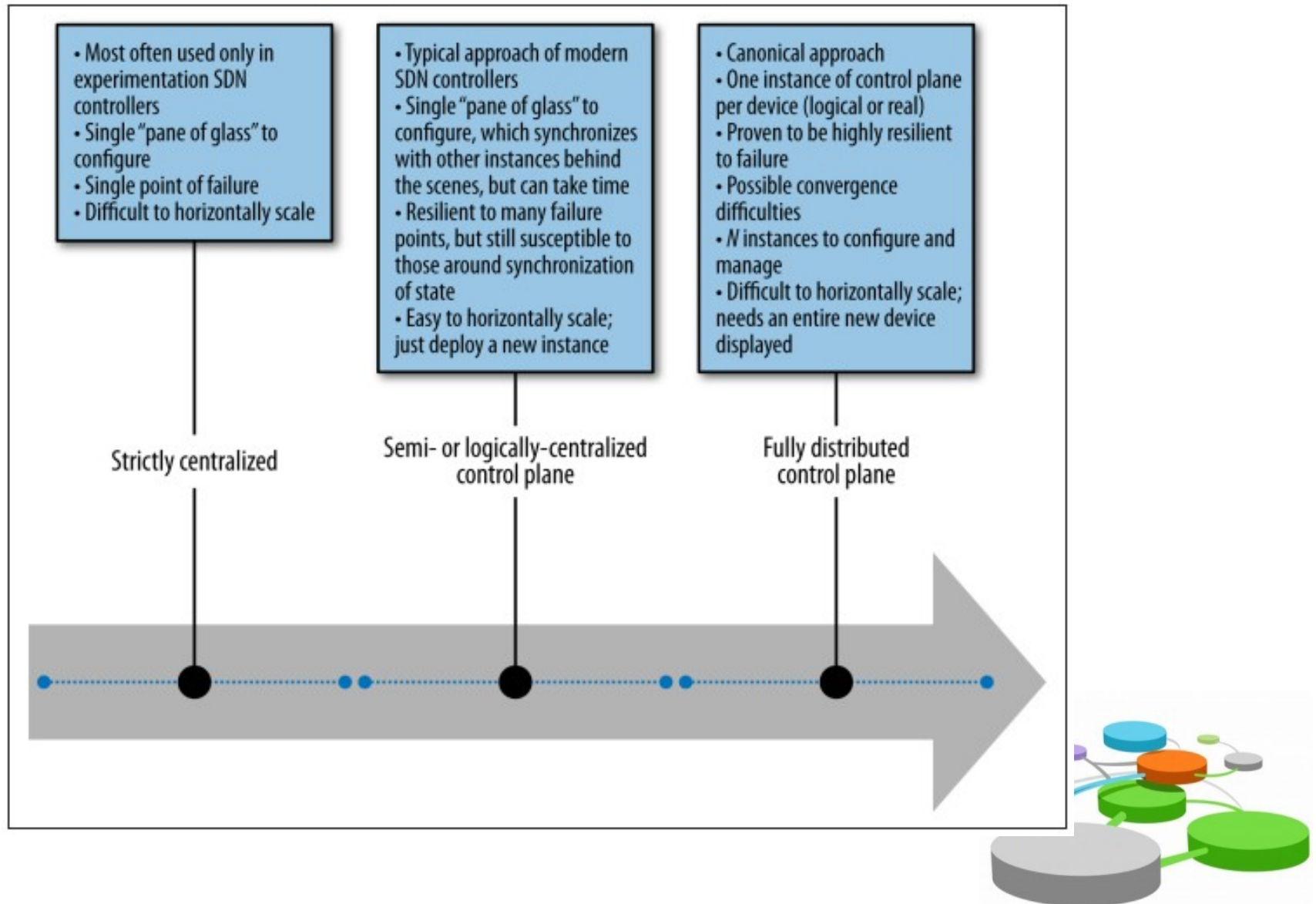
(IETF RFC 7426)

“The physical separation of the network control plane from the forwarding plane, and where a control plane controls several devices.”

(ONF – Open Networking Foundation)



Software-Defined Networking



Software-Defined Networking

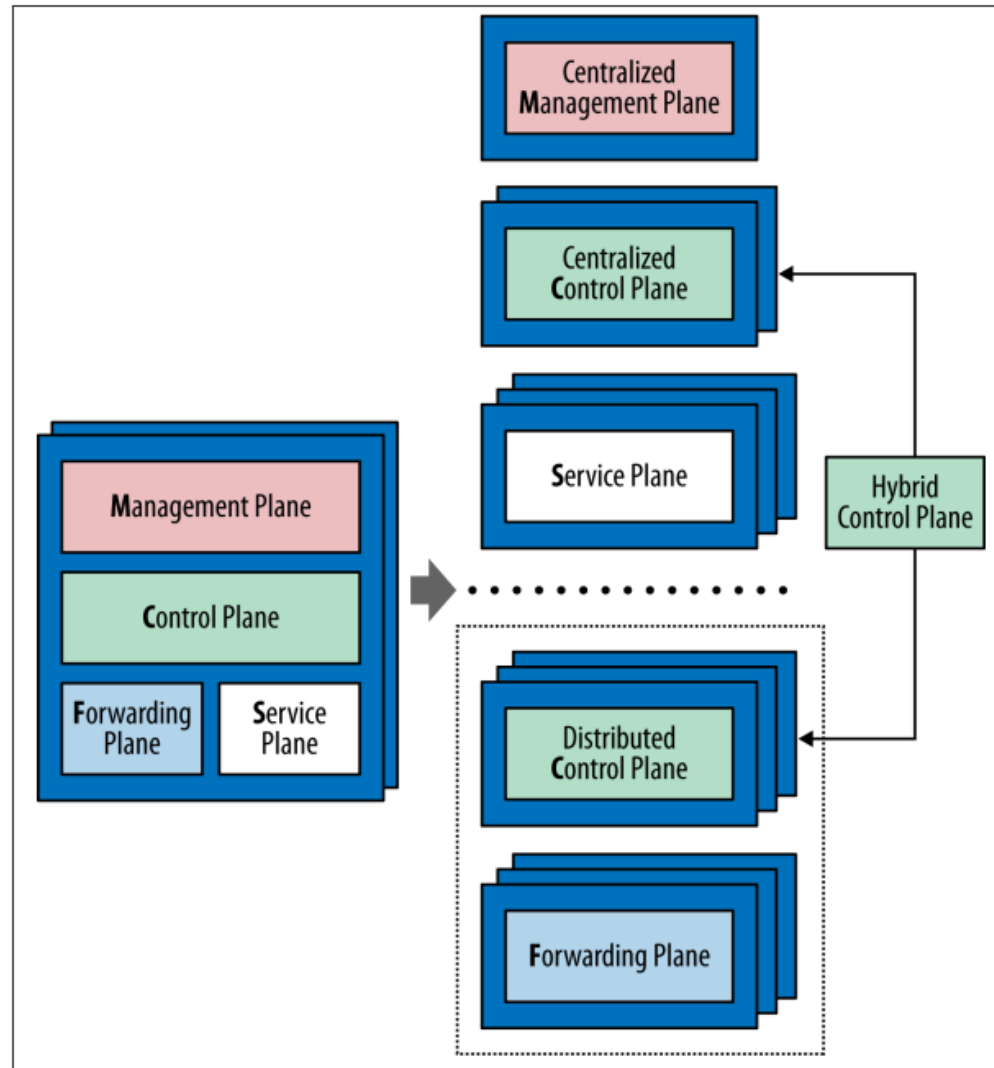


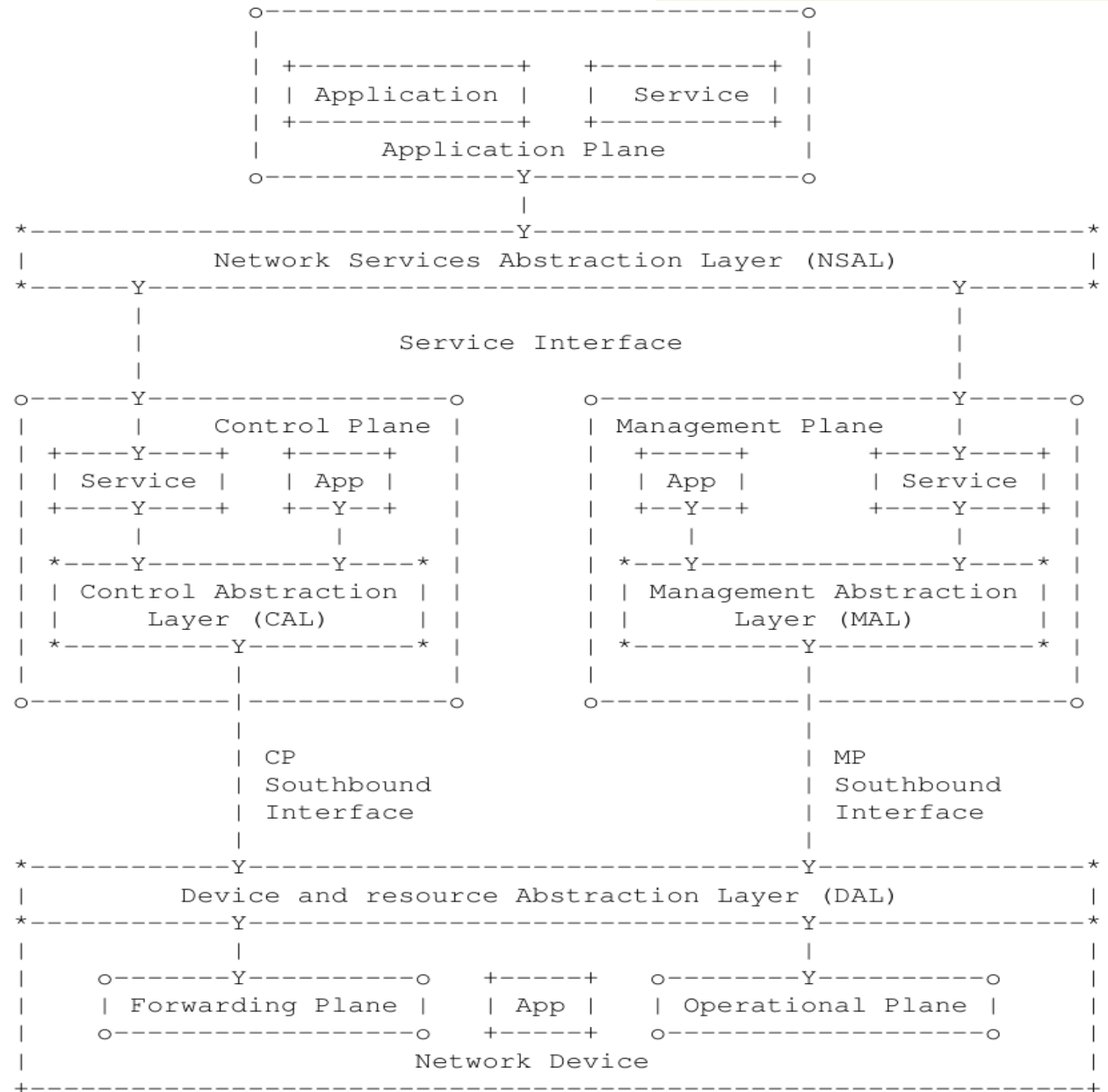
Figure 2-7. Separating the integrated management, control, service, and forwarding planes so that they can scale independently



Software-Defined Networking

Arquitectura
SDN

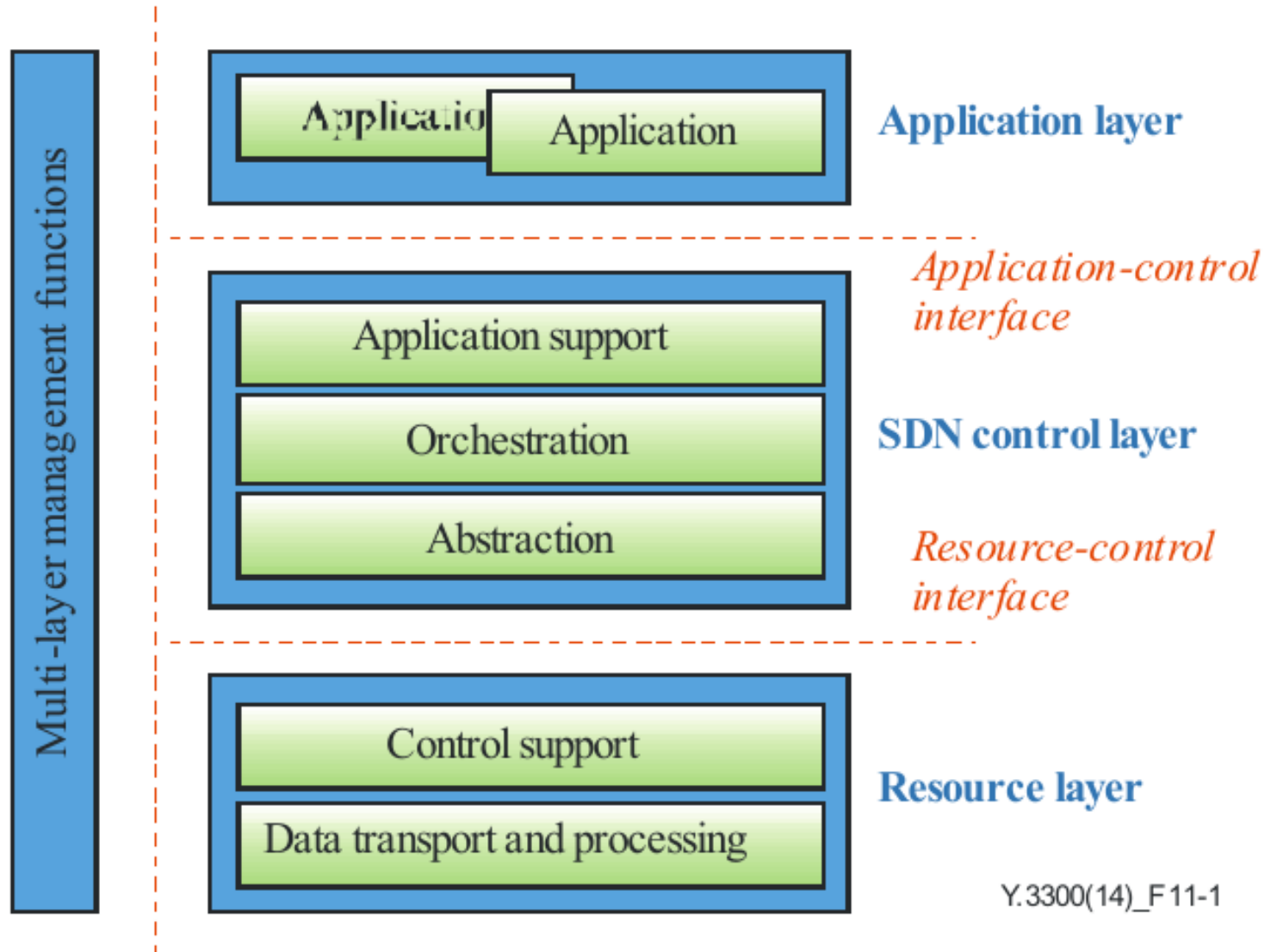
IETF - RFC
7426



Software-Defined Networking

Arquitectura
SDN

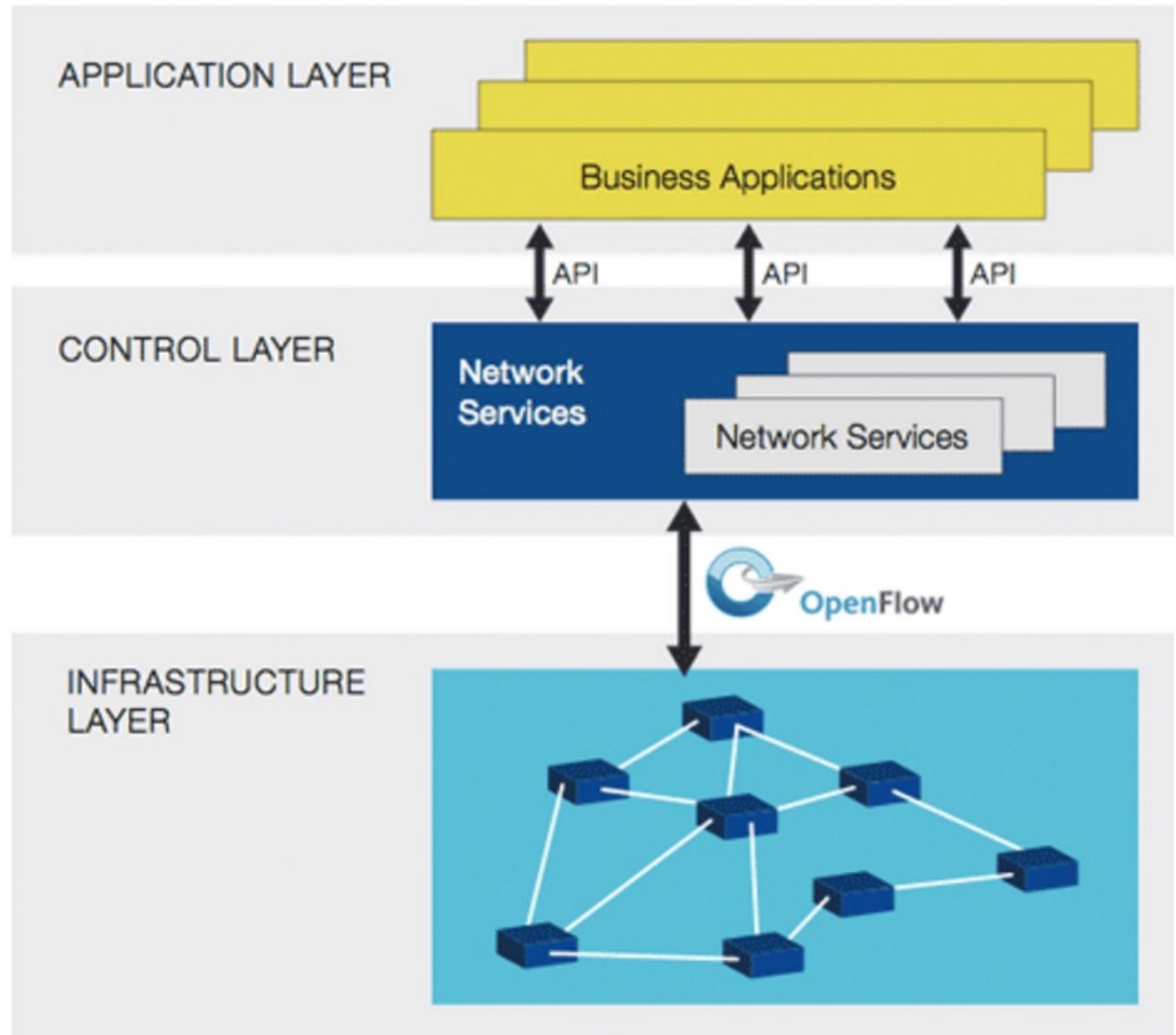
UIT-T Y.3300



Software-Defined Networking

Arquitectura
SDN

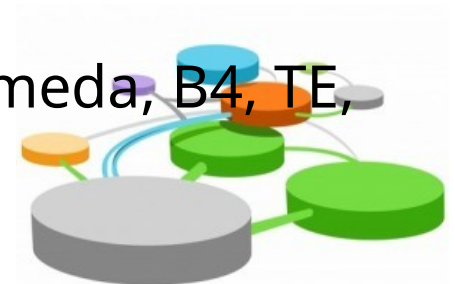
ONF



Software-Defined Networking

Actualmente diversos controladores/plataformas

- Open source:
 - OpenDaylight, Tungsten Fabric (antes OpenContrail), Floodlight, Onos, Ryu, Beacon, Nox/Pox
 - ONAP: merge de open source ECOMP (liberado por AT&T en 2017) y Open Orchestrator Project (OPEN-O)
- Proprietarios:
 - Brocade, Cisco, Big Switch Networks, Dell, HPE, Cumulus Network, Juniper, IBM, NEC, VMWare, Nuage , Pica8...
- Internos:
 - Google (Al menos 5 controladores: FirePath, Andromeda, B4, TE, BwE, uso de openflow), Microsoft ...



Software-Defined Networking

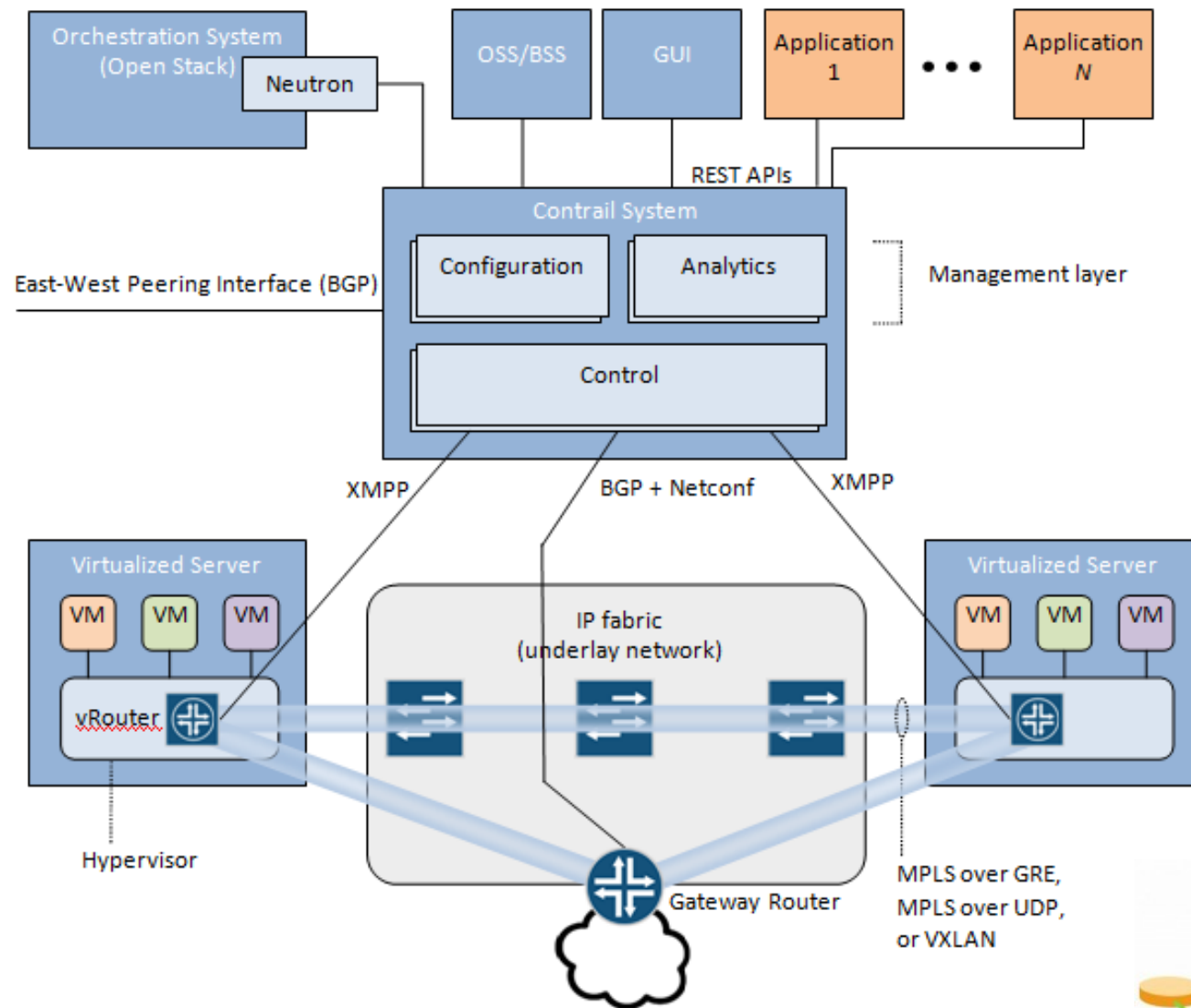


Figure 1: OpenContrail System Overview



Software-Defined Networking

SDN en Carriers Globales

(AT&T, Level3, Colt, Orange BS, SK Telecom, Telefónica ...)

● Encuesta tiempo de despliegue de IHS (2018):

- 78 % de encuestados dice haber implementado o implementarán SDN en 2018
- 9% en 2019.
- 4% en 2020 o luego.
- 9% En algún momento harán despliegue...

● En AT&T:

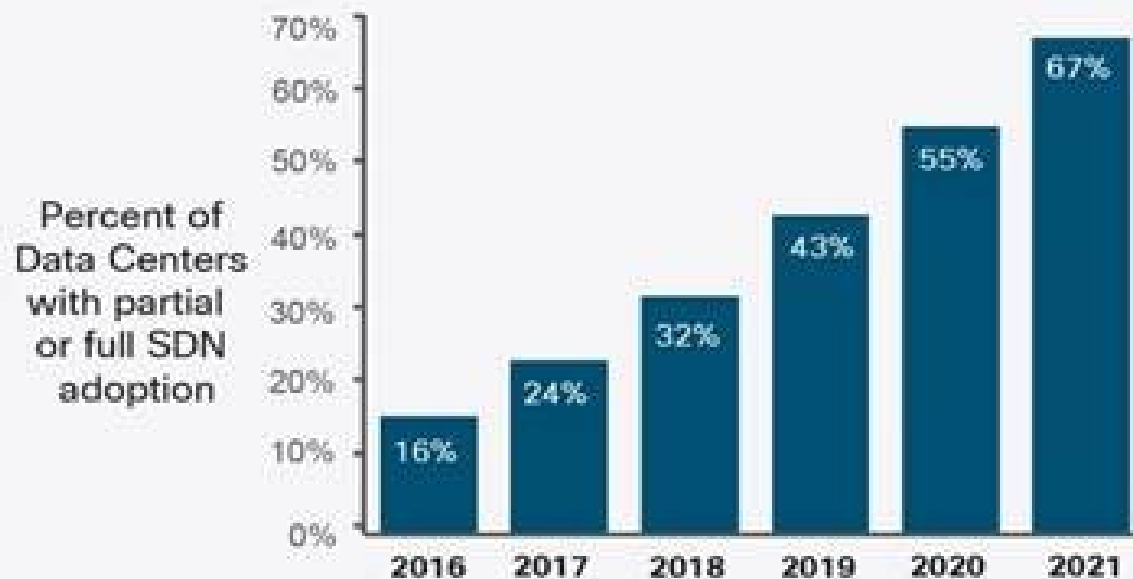
- Virtualización en 2015 5.7%, planeaban 30% para 2016 y 75% para 2020.
- Septiembre 2019 [1]: 75% del tráfico sobre túneles MPLS, que conectan
- los elementos de “core” de su red están bajo el control SDN.

[1] https://about.att.com/innovationblog/2019/09/delivering_sdn_promise.html



Software-Defined Networking

Tráfico SDN en Datacenters



Source: Cisco Global Cloud Index, 2016-2021.

Software-Defined Networking

Tráfico SDN en Datacenters



Source: Cisco Global Cloud Index, 2016-2021.

Software-Defined Networking

Propuesta OpenFlow:

- 2008: UC Berkeley, Washington, Princeton “OpenFlow: Enabling Innovation in Campus Networks”
- Aprovechar las funciones comunes ya implementadas en flow-tables.

OpenFlow Switch Specification (Open Network Foundation)

Version actual* 1.5.1 – Marzo 2015:

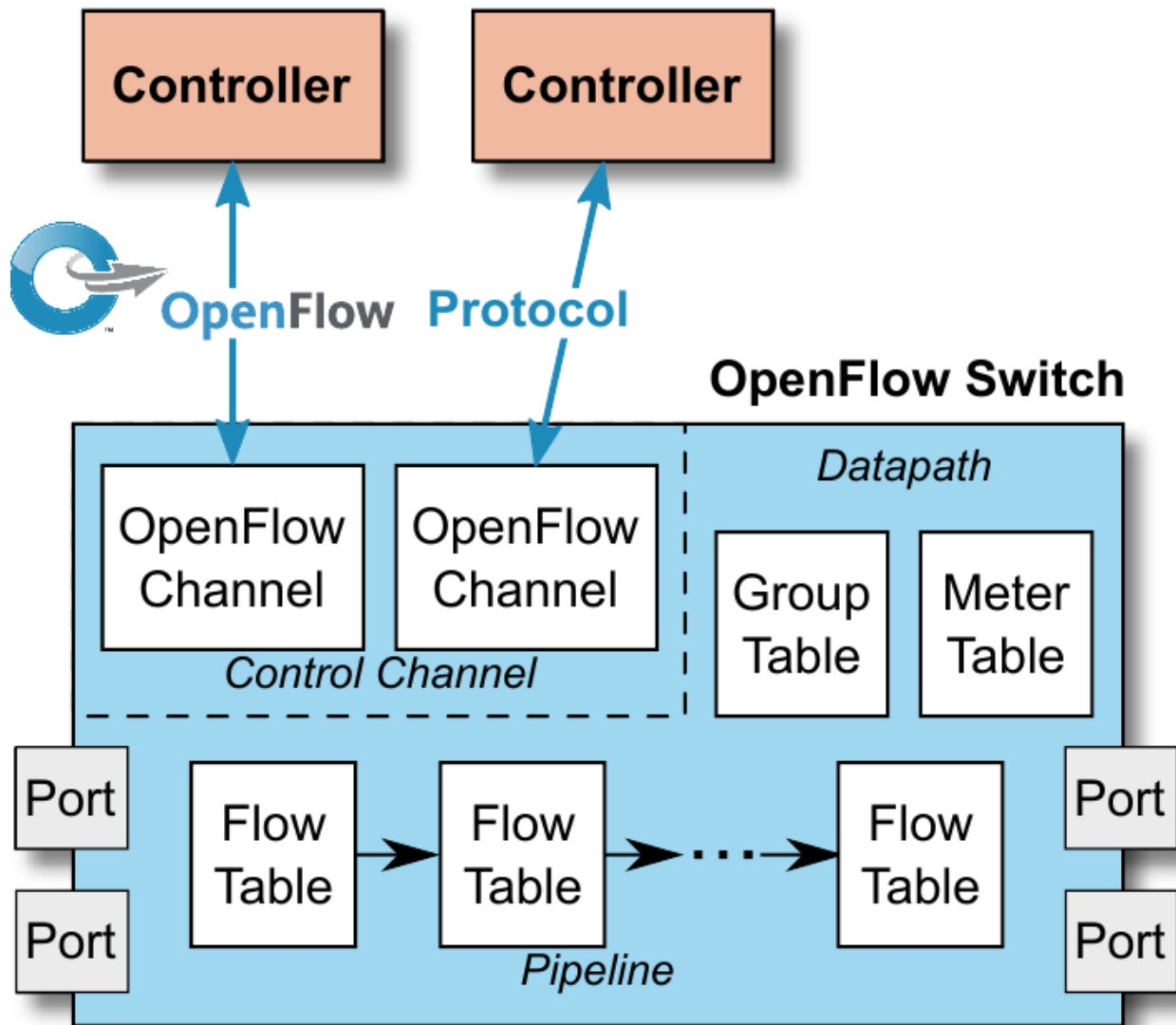
- Openflow Switch:
 - Flow-tables
 - Canal seguro de comunicación con el controlador
 - El protocolo OpenFlow (interfaz estándar para definir las flow-tables)
- Switchs dedicados o híbridos.

* Versión 1.6 (2016-09) Members only



Software-Defined Networking

OpenFlow Switch – Componentes principales



Software-Defined Networking

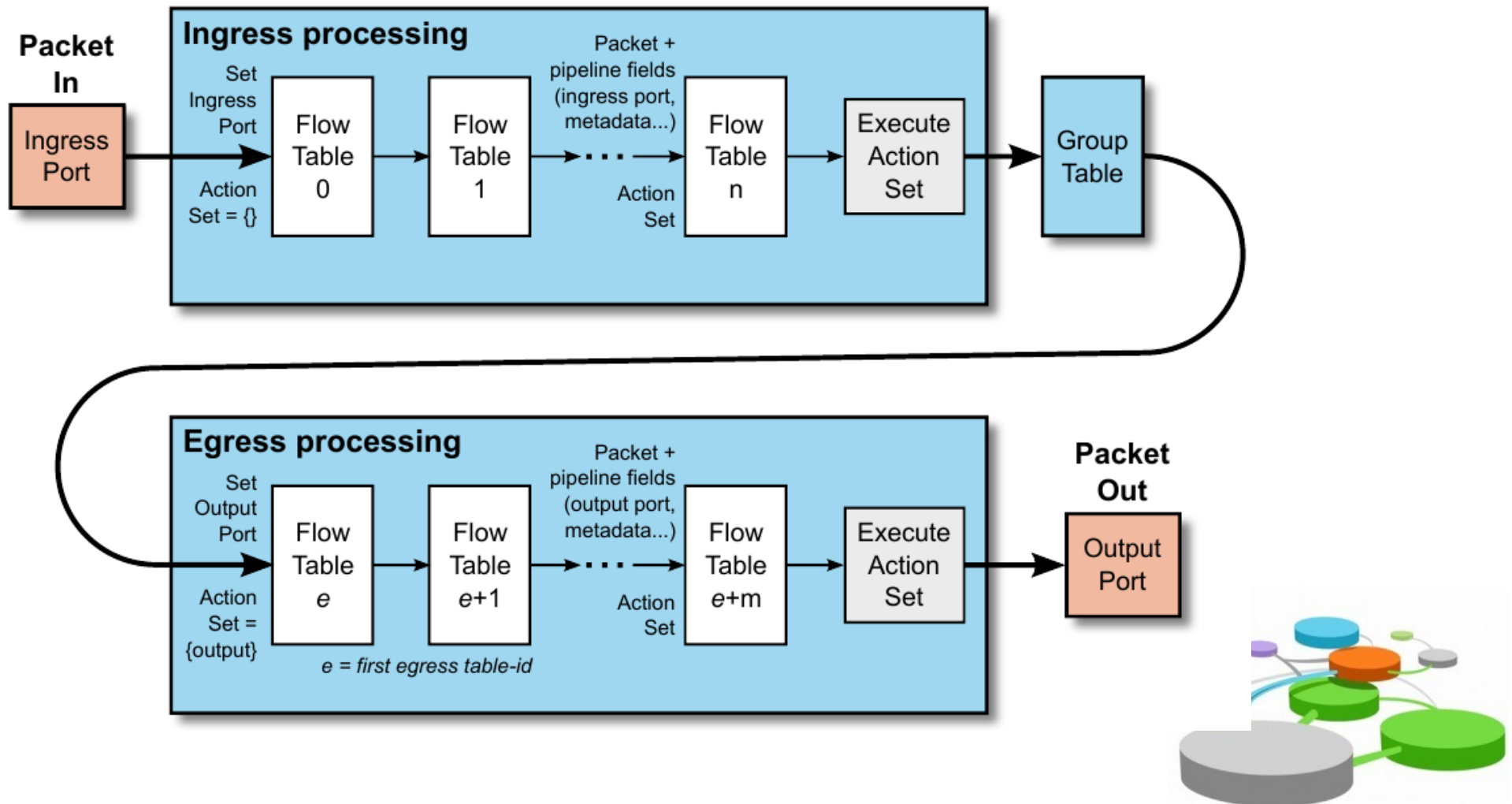
OpenFlow Switch Specification

- El controlador agrega, modifica o borra entradas de las flow tables utilizando el protocolo OpenFlow switch protocol.
- Las actualizaciones pueden ser proactivas o reactivas.
- Cada entrada en la Flow Table contiene:
 - Campos de matcheo
 - Acciones (conjunto de instrucciones)
 - Contadores



Software-Defined Networking

OpenFlow Pipeline Processing



Software-Defined Networking

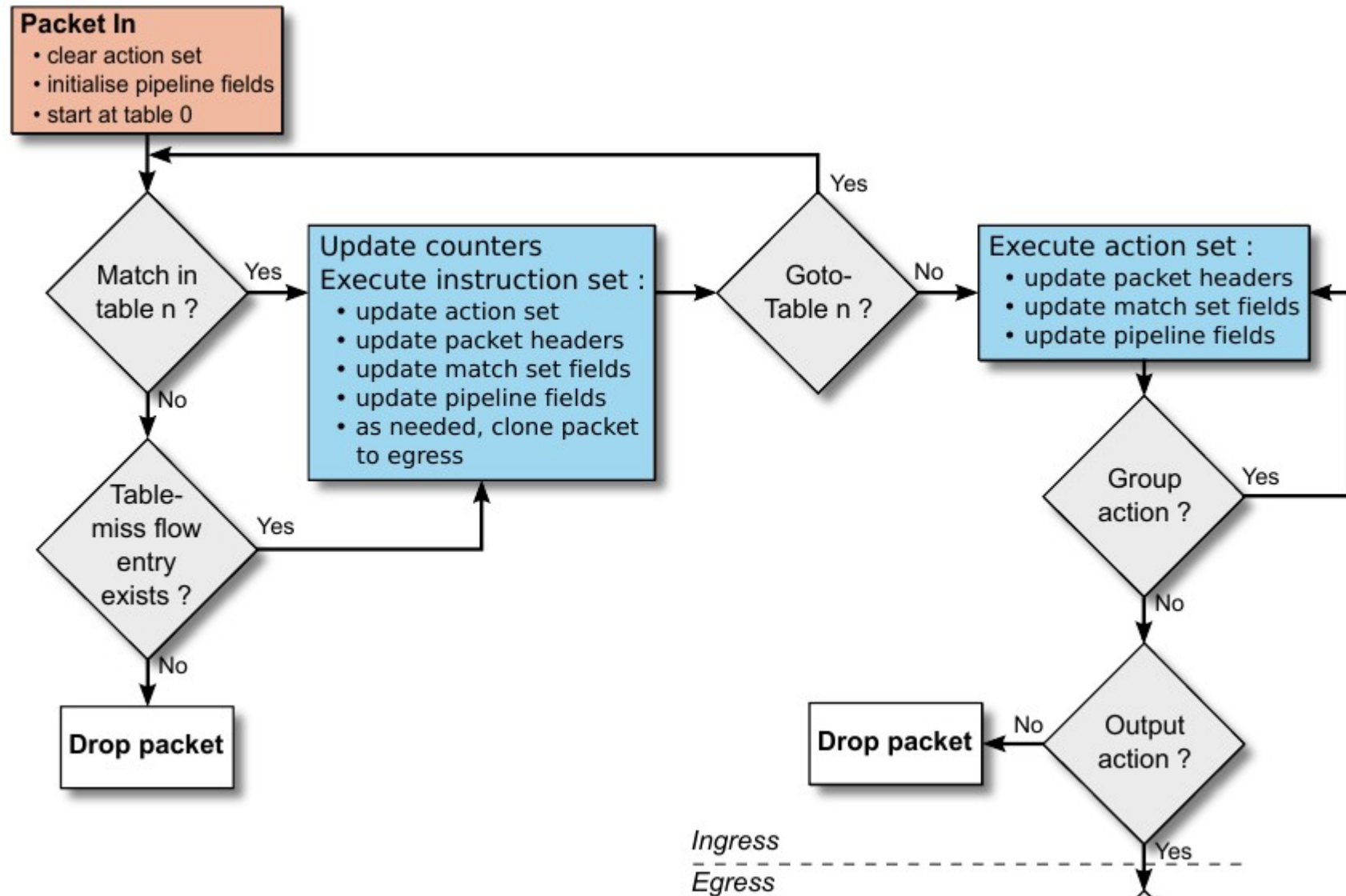
OpenFlow Switch Specification Flow Table Entry

Match Fields	Priority	Counters	Instructions	Timeouts	Cookie	Flags
--------------	----------	----------	--------------	----------	--------	-------

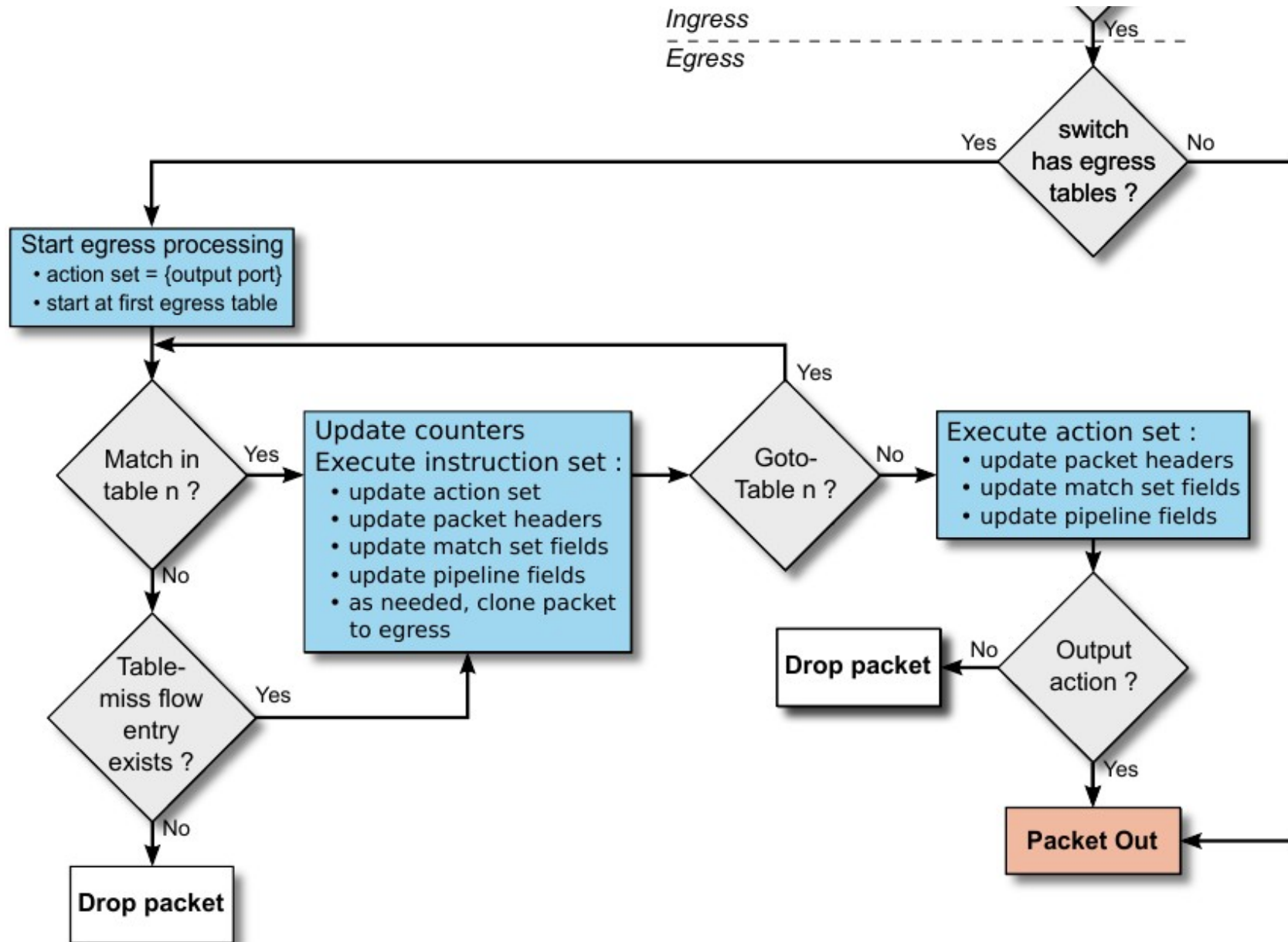
- **match fields:** Campos de “matcheo”: Puerto de ingreso y encabezados del paquete, y opcionalmente otros datos del procesamiento tal como metadatos agregados por tablas anteriores.
- **priority:** precedencia de la entrada.
- **counters:** actualizados cuando matchean paquetes.
- **instructions:** modifican el conjunto de acciones (action set) o el procesamiento en tubería
- **timeouts:** Tiempo máximo de la entrada o tiempo máximo sin actividad.
- **cookie:** Valor colocado por el controlador. Puede ser utilizado para filtrar entradas afectadas por estadísticas, requerimientos de modificación o borrado. No utilizado en el procesamiento de paquetes.
- **flags:** Determinan como son administradas las entradas.



Software-Defined Networking



Software-Defined Networking



Software-Defined Networking

Ports

- Físicos
- Lógicos (tuneles, loopback, grupos de agregación de enlaces)
- Reservados:
 - All
 - Controller
 - Table
 - IN_PORT
 - ANY
 - UNSET
 - Local (Opcional)
 - Normal (Opcional)
 - Flood (Opcional)



Software-Defined Networking

Instrucciones

- Requeridas:
 - Write-Actions: Agrega acción/es al conjunto de acciones actual.
 - Goto-Table: Indica la próxima tabla a procesar.
 - Clear-Actions: Vaciar el Action Set.
- Opcionales:
 - Apply-Actions: Aplica las acciones especificadas inmediatamente sin afectar el action set.
 - Stat-Trigger: Generar un evento al controlador si alguna estadística del flujo atraviesa un umbral.
 - Write-Metadata: Modifica el campo de metadatos.



Software-Defined Networking

Acciones

- Requeridas:
 - Output *port_no.*: Forwarda el paquete por el puerto Open Flow especificado donde comienza el egress processing.
 - Group *group_id.*: Procesa el paquete por el grupo especificado.
 - Drop.
- Opcionales:
 - Set-Queue *queue_id.*: Utilizado para determinar en cuál cola del puerto se encolará el paquete.
 - Meter *meter_id.*: Dirigir paquete al “meter” especificado.



Software-Defined Networking

Acciones (2)

- Opcionales:
 - Push-Tag/Pop-Tag *ethertype*: Push o Pop de encabezados VLAN, MPLS o PBB.
 - Set-Field *field_type value*: Modifica valor de campo de encabezado del paquete.
 - Copy-Field *src_field_type dst_field_type*: Copiar valores entre headers o campos del pipeline.
 - Change-TTL *tvl*: Modifica IPv4 TTL, IPv6 Hop Limit o MPLS TTL.



Software-Defined Networking

OpenFlow Switch Specification Match Fields v1.3.4

Switch input port.
Switch physical input port.
Metadata passed between tables.
Ethernet destination address.
Ethernet source address.
Ethernet frame type.
VLAN id.
VLAN priority.
IP DSCP (6 bits in ToS field).
IP ECN (2 bits in ToS field).
IP protocol.
IPv4 source address.
IPv4 destination address.
TCP source port.
TCP destination port.
UDP source port.
UDP destination port.
SCTP source port.
SCTP destination port.
ICMP type.

ICMP code.
ARP opcode.
ARP source IPv4 address.
ARP target IPv4 address.
ARP source hardware address.
ARP target hardware address.
IPv6 source address.
IPv6 destination address.
IPv6 Flow Label.
ICMPv6 type.
ICMPv6 code.
Target address for ND.
Source link-layer for ND.
Target link-layer for ND.
MPLS label.
MPLS TC.
MPLS BoS bit.
PBB I-SID.
Logical Port Metadata.
IPv6 Extension Header pseudo-field



Software-Defined Networking

Mensajes (TCP/TLS port 6653)

Tres tipos de mensajes

- **Controller-to-Switch:** Iniciados por el controller
 - Administrar o inspeccionar el estado del switch
 - Puede o no requerir respuesta del switch
 - Features, Configuration, Modify-state, Read-state, Packet-out, Barrier, Role-request, Asynchronous-configuration
- **Asynchronous**
 - Enviados por el switch sin que lo solicite el controlador
 - Informan de paquetes que arriban, cambios en el estado del switch, errores.
 - Packet-in, Flow-removed, Port-status, Error
- **Symmetric**
 - Enviados por switch o controlador sin ser solicitados
 - Hello, Echo, Experimenter



Software-Defined Networking

Otras características

- **Múltiples controladores:**

- Role equal:

- Pueden enviar mensajes controller-to-switch

- Reciben todos los mensajes asíncronos.

- Role Slave:

- read-only access sobre el switch.

- Por defecto no reciben todos los mensajes asíncronos, salvo los port-status

- Role Master

- Similar a equal, solamente uno en ese estado.

- **Conexiones auxiliares**

- Opcionalmente creadas por el switch para performance aprovechando sus capacidades-paralelismo (TLS, TCP, DTLS o UDP)

Software-Defined Networking

OpenFlow: enabling innovation in campus networks. Nick McKeown, Tom Anderson, HariBalakrishnan, Guru Parulkar, Larry Peterson, Jennifer RexfordScott Shenker, Jonathan Turner. ACM SIGCOMM Computer Communication Review, Volume 38, Number 2, April 2008.

Software-Defined Networking: The New Norm for Networks. ONF White paper. 2012

<http://www.opennetsummit.org/archives/apr12/hoelzle-tue-openflow.pdf>

SDN: Software Defined Networks An Authoritative Review of Network Programmability Technologies. O'Reilly Media. 2013

<https://www.opennetworking.org/>

RFC 7426 “Software-Defined Networking (SDN): Layers and Architecture Terminology”

UIT-T Rec Y.3300 “Framework of software-defined networking”