

# Proyecto de grado: Routers Reconfigurables de Altas Prestaciones

Rodrigo Amaro, Emiliano Viotti

Instituto de Computación  
Facultad de Ingeniería  
Universidad de la República

Tutores: Dr. Eduardo Grampín, MSc. Martín Giachino

18 de agosto de 2015

# Agenda

- 1 Introducción
- 2 Conceptos preliminares
- 3 Arquitectura propuesta
- 4 Implementación
- 5 Experimentación
- 6 Conclusiones
- 7 Trabajo a futuro

## 1 Introducción

2 Conceptos preliminares

3 Arquitectura propuesta

4 Implementación

5 Experimentación

6 Conclusiones

7 Trabajo a futuro

# Motivación I

## Redes académicas

Internet no parece apropiada para su utilización en el contexto académico en actividades de enseñanza, investigación, el desarrollo de nuevos protocolos, servicios e innovación en el área.



## Motivación II

### Red Académica Uruguaya (RAU)

A nivel local, la RAU es un emprendimiento de la Universidad de la República administrado por el SeCIU con los objetivos de unir a las instituciones académicas nacionales en una red de alcance nacional y a través de ella conectarlas a Latinoamérica.

### RAU2

Remplazo de la actual red académica, es una red avanzada de altas prestaciones que estaría dotada de funciones de virtualización de redes flexibles en su definición y uso.



## Hardware comercial

Los equipos de red de backbone comerciales como HP, CISCO, Juniper son costos y generalmente de naturaleza cerrada. Las funcionalidades del hardware se restringen a las funcionalidades expuesta por una API propietaria.



# Definición del problema I

## Definición del Problema

Construir un prototipo para la RAU2 utilizando como plataforma PCs con placas de red aceleradas en hardware reconfigurable NetFPGA y el enfoque de las Redes Definidas por Software (SDN).

# Definición del problema I

## Definición del Problema

Construir un prototipo para la RAU2 utilizando como plataforma PCs con placas de red aceleradas en hardware reconfigurable NetFPGA y el enfoque de las Redes Definidas por Software (SDN).

## Resultados esperados

- Estado del arte en Redes Definidas por Software y hardware NetFPGA

# Definición del problema I

## Definición del Problema

Construir un prototipo para la RAU2 utilizando como plataforma PCs con placas de red aceleradas en hardware reconfigurable NetFPGA y el enfoque de las Redes Definidas por Software (SDN).

## Resultados esperados

- Estado del arte en Redes Definidas por Software y hardware NetFPGA
- Prototipo de aplicación de gestión de red utilizando SDN y el hardware NetFPGA (orientado a los requerimientos de la RAU2)

# Definición del problema I

## Definición del Problema

Construir un prototipo para la RAU2 utilizando como plataforma PCs con placas de red aceleradas en hardware reconfigurable NetFPGA y el enfoque de las Redes Definidas por Software (SDN).

## Resultados esperados

- Estado del arte en Redes Definidas por Software y hardware NetFPGA
- Prototipo de aplicación de gestión de red utilizando SDN y el hardware NetFPGA (orientado a los requerimientos de la RAU2)
- Diseño e implementación de pruebas de verificación

# Definición del problema II

¿Qué características se esperan del prototipo, cuáles son los requerimientos?

## Definición del problema II

¿Qué características se esperan del prototipo, cuáles son los requerimientos?

R: Pensemos en posibles requerimientos sobre la RAU2

# Definición del problema II

¿Qué características se esperan del prototipo, cuáles son los requerimientos?

R: Pensemos en posibles requerimientos sobre la RAU2

## Requerimientos

- ① Clasificación y separación de tráfico según categorías

# Definición del problema II

¿Qué características se esperan del prototipo, cuáles son los requerimientos?

R: Pensemos en posibles requerimientos sobre la RAU2

## Requerimientos

- ① Clasificación y separación de tráfico según categorías
- ② Manejo de grandes volúmenes de datos

# Definición del problema II

¿Qué características se esperan del prototipo, cuáles son los requerimientos?

R: Pensemos en posibles requerimientos sobre la RAU2

## Requerimientos

- ① Clasificación y separación de tráfico según categorías
- ② Manejo de grandes volúmenes de datos
- ③ Escalabilidad

# Definición del problema II

¿Qué características se esperan del prototipo, cuáles son los requerimientos?

R: Pensemos en posibles requerimientos sobre la RAU2

## Requerimientos

- ① Clasificación y separación de tráfico según categorías
- ② Manejo de grandes volúmenes de datos
- ③ Escalabilidad

# Definición del problema II

¿Qué características se esperan del prototipo, cuáles son los requerimientos?

R: Pensemos en posibles requerimientos sobre la RAU2

## Requerimientos

- ① Clasificación y separación de tráfico según categorías
- ② Manejo de grandes volúmenes de datos
- ③ Escalabilidad

Problema demasiado grande!

# Definición del problema II

¿Qué características se esperan del prototipo, cuáles son los requerimientos?

R: Pensemos en posibles requerimientos sobre la RAU2

## Requerimientos

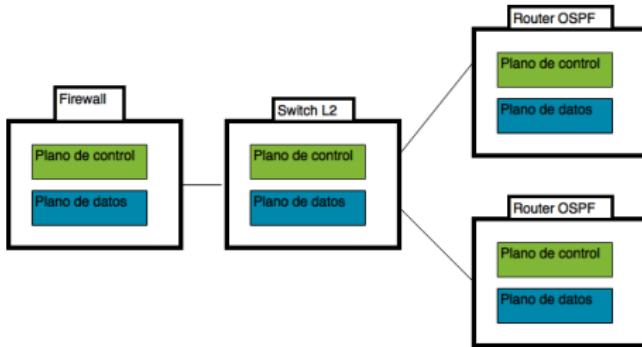
- ① Clasificación y separación de tráfico según categorías
- ② Manejo de grandes volúmenes de datos
- ③ Escalabilidad

Problema demasiado grande!

Se hace foco en resolver (1)

- 1 Introducción
- 2 Conceptos preliminares
- 3 Arquitectura propuesta
- 4 Implementación
- 5 Experimentación
- 6 Conclusiones
- 7 Trabajo a futuro

## Enfoque tradicional de redes



## Plano de control

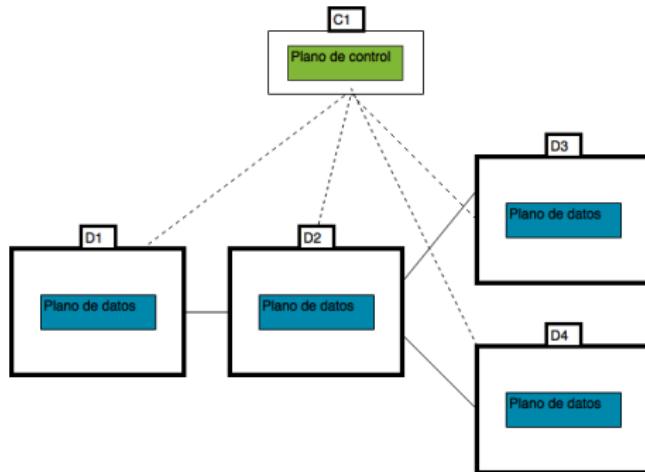
Es donde reside la inteligencia de la red, es la lógica que controla el comportamiento de reenvío. Ejs : OSPF, RIP, Firewalls

## Plano de datos

Se encarga de reenviar el tráfico conforme a el plano de control Ejs : Reenvío IP

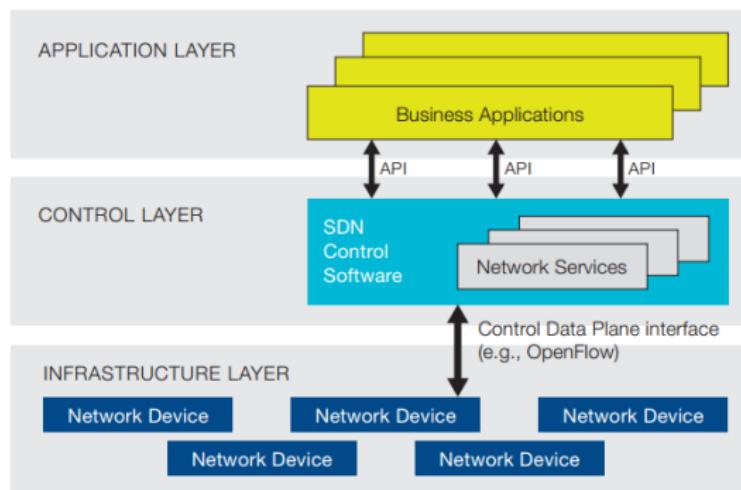
## Definicion

La separación física del plano de control de la red del plano de datos, donde el plano de control controla varios dispositivos.



# SDN (Capas lógicas)

- Tres capas: (1) capa de aplicaciones, (2) capa de control y (3) capa de infraestructura
- Dos interfaces de comunicación: interfaz sur e interfaz norte



# OpenFlow

Solución basada en el enfoque SDN, ofrece una implementación estándar de la Interfaz Sur. El protocolo especifica:

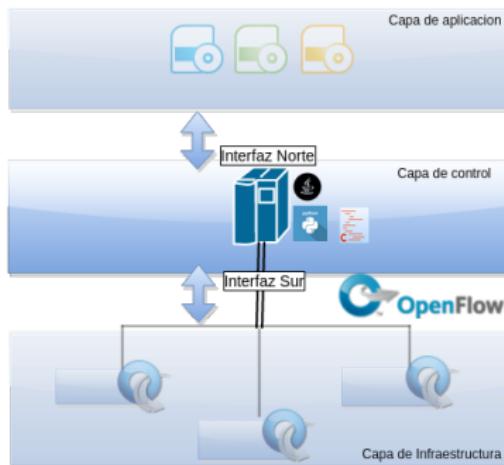
- Canal seguro de comunicación entre capa de control e infraestructura (Protocolo OpenFlow)
- Especificación de dispositivos compatibles con la arquitectura (Switch OpenFlow)

# OpenFlow (Capa de Aplicación)



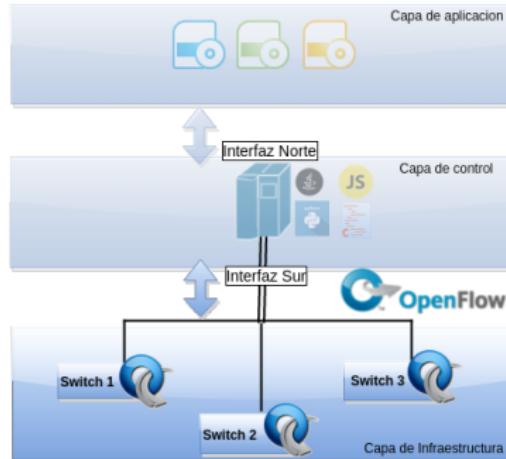
- El comportamiento de la red se implementa mediante aplicaciones
- Se escriben en lenguajes de programación y de alto nivel (C/C++, Java, Python)
- Se ejecutan sobre el Controlador OpenFlow y hacen uso de la API Norte

# OpenFlow (Capa de Control)



- Implementación en Software que se ejecuta en hardware convencional (x86)
- Ofrece una API de alto nivel para controlar los dispositivos de una Red (Interfaz Norte)
- Cada operación se traduce a reglas OpenFlow y comunicadas al dispositivo a través de la Interfaz Sur (protocolo OpenFlow)

# OpenFlow (Capa de Dispositivos)



Cada dispositivo OpenFlow implementa  
**Tablas de Flujos**



- ① **Regla:** Define una clase de tráfico
- ② **Acción:** Define el procesamiento (Drop, Output, Flood, GoToTable)
- ③ **Estadísticas:** Cantidad de paquetes procesados por cada flujo

# VPN (Red Privada Virtual)

Extensión de una red privada sobre la infraestructura de una red pública.  
Nos interesa diferenciarlas de acuerdo a:

- Que capa OSI virtualizan: L2 o L3
- Conexión utilizada: punto a punto o multipunto



# MPLS

MPLS (Multilabel Protocol Switching) es la solución de facto para la implementación de servicios de VPN L2 y L3. Permite transportar tráfico mediante la conmutación de etiquetas.

MPLS (Multilabel Protocol Switching) es la solución de facto para la implementación de servicios de VPN L2 y L3. Permite transportar tráfico mediante la conmutación de etiquetas.

Arquitectura:

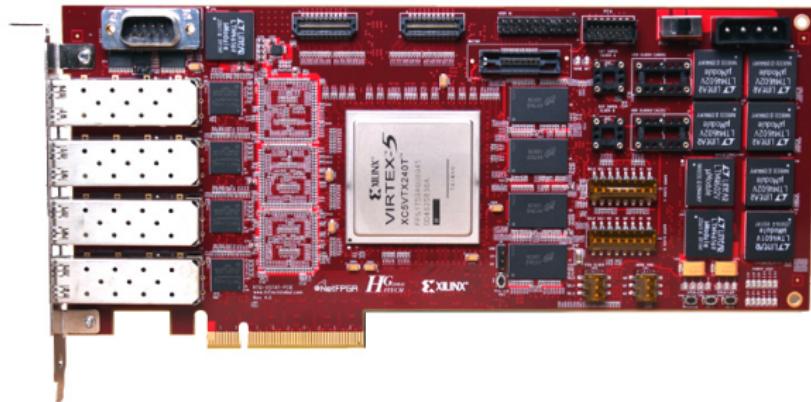
- Label: Etiqueta MPLS (20bits)
- Acción: Pop, Push y Swap
- FEC: Clase de equivalencia de tráfico
- LSP: Camino de conmutación de etiquetas
- LDP: Protocolo de distribución de etiquetas

# NetFPGA

Plataforma de hardware de red reconfigurable y software Open Source.

Hardware: Placa PCI-E que cuenta con:

- ① Chip reconfigurable FPGA
- ② 27Mb SRAM + 288MBytes RLDRAM
- ③ 4 Puertos 10-Gigabit SFP+ Ethernet



## Software

Proyectos de software Open Source que permiten programar comportamientos en el hardware. Dos tipos de proyectos: Referencia y Comunitarios

Proyecto <sup>1</sup>	Organización
Production Test	Stanford University
Learning CAM Switch	Stanford University/University of Cambridge
Reference Router	Stanford University/University of Cambridge
Reference NIC 10G	Stanford University/University of Cambridge
OpenFlow Switch	Stanford University

<sup>1</sup> Proyectos NetFPGA extraídos de <http://netfpga.org/site/#/systems/3netfpga-10g/applications/>

- 1 Introducción
- 2 Conceptos preliminares
- 3 Arquitectura propuesta
- 4 Implementación
- 5 Experimentación
- 6 Conclusiones
- 7 Trabajo a futuro

# Arquitectura propuesta I

- ¿Cómo implementamos clasificación de tráfico y separamos en categorías?

# Arquitectura propuesta I

- ¿Cómo implementamos clasificación de tráfico y separamos en categorías?

R: Construyendo Redes Privadas Virtuales (VPNs)

# Arquitectura propuesta I

- ¿Cómo implementamos clasificación de tráfico y separamos en categorías?

R: Construyendo Redes Privadas Virtuales (VPNs)

- ¿Cómo implementamos Redes Privadas Virtuales?

# Arquitectura propuesta I

- ¿Cómo implementamos clasificación de tráfico y separamos en categorías?

R: Construyendo Redes Privadas Virtuales (VPNs)

- ¿Cómo implementamos Redes Privadas Virtuales?

R: Utilizando MPLS (Multilabel Protocol Switch)

# Arquitectura propuesta I

- ¿Cómo implementamos clasificación de tráfico y separamos en categorías?

R: Construyendo Redes Privadas Virtuales (VPNs)

- ¿Cómo implementamos Redes Privadas Virtuales?

R: Utilizando MPLS (Multilabel Protocol Switch)

- ¿VPNs con MPLS en SDN?

# Arquitectura propuesta I

- ¿Cómo implementamos clasificación de tráfico y separamos en categorías?

R: Construyendo Redes Privadas Virtuales (VPNs)

- ¿Cómo implementamos Redes Privadas Virtuales?

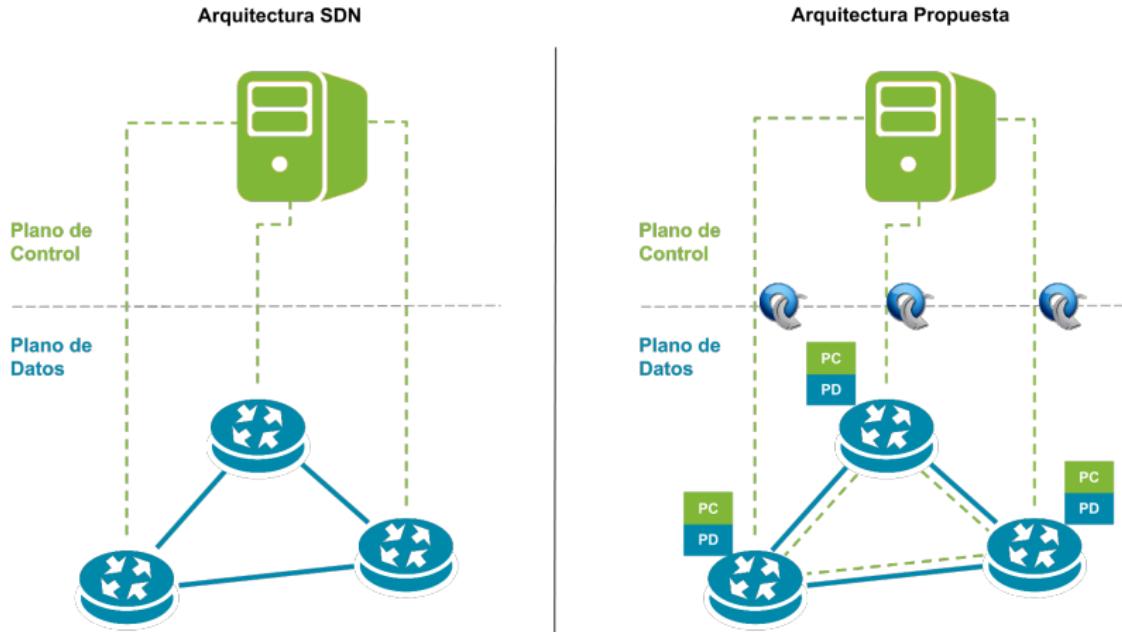
R: Utilizando MPLS (Multilabel Protocol Switch)

- ¿VPNs con MPLS en SDN?

R: OpenFlow!

# Arquitectura propuesta III

Se propone una arquitectura híbrida que combina el enfoque SDN con tecnologías actuales, basándose en la implementación de OpenFlow.



# Switch OpenFlow

Se identifican dos alternativas utilizando hardware NetFPGA:

# Switch OpenFlow

Se identifican dos alternativas utilizando hardware NetFPGA:

- ① Proyecto OpenFlow

# Switch OpenFlow

Se identifican dos alternativas utilizando hardware NetFPGA:

- ① Proyecto OpenFlow
- ② Proyecto ReferenceNIC más implementación OF en software (Open vSwitch)

# Switch OpenFlow

Se identifican dos alternativas utilizando hardware NetFPGA:

- ① Proyecto OpenFlow
- ② Proyecto ReferenceNIC más implementación OF en software (Open vSwitch)

(1) OpenFlow	(2) ReferenceNIC y OvS
Velocidad de procesamiento	Prototipación agil
Utilización del hardware	Tiempo libre para investigación
Conocimiento específico	No hay que utilizar HDLs
Licencias full	Licencias económicas

# Switch OpenFlow

Se identifican dos alternativas utilizando hardware NetFPGA:

- ① Proyecto OpenFlow
- ② Proyecto ReferenceNIC más implementación OF en software (Open vSwitch)

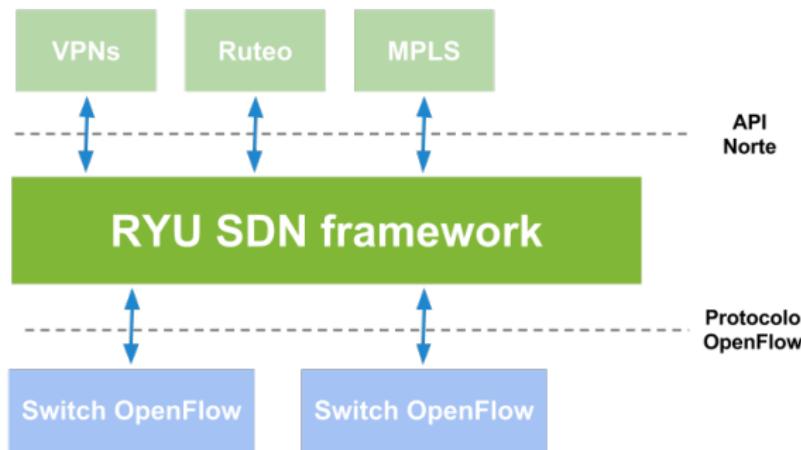
(1) OpenFlow	(2) ReferenceNIC y OvS
Velocidad de procesamiento	Prototipación agil
Utilización del hardware	Tiempo libre para investigación
Conocimiento específico	No hay que utilizar HDLs
Licencias full	Licencias económicas

## (2) ReferenceNIC y OvS!

# Controlador OpenFlow

Utilizamos Ryu como software de control SDN

- Compatible con OF hasta v1.4
- Académico, sencillo, liviano
- Libre y Open Source



# Algoritmo de Ruteo

- Calculo de rutas en base a IP
- Utilizamos OSPF para la construcción de una base de datos topológica (LSDB)
- Cada nodo en la red incluyendo el controlador ejecutan un demonio OSPF de la herramienta Quagga
- Algoritmo de ruteo como App SDN en el controlador, toma información de la LSDB

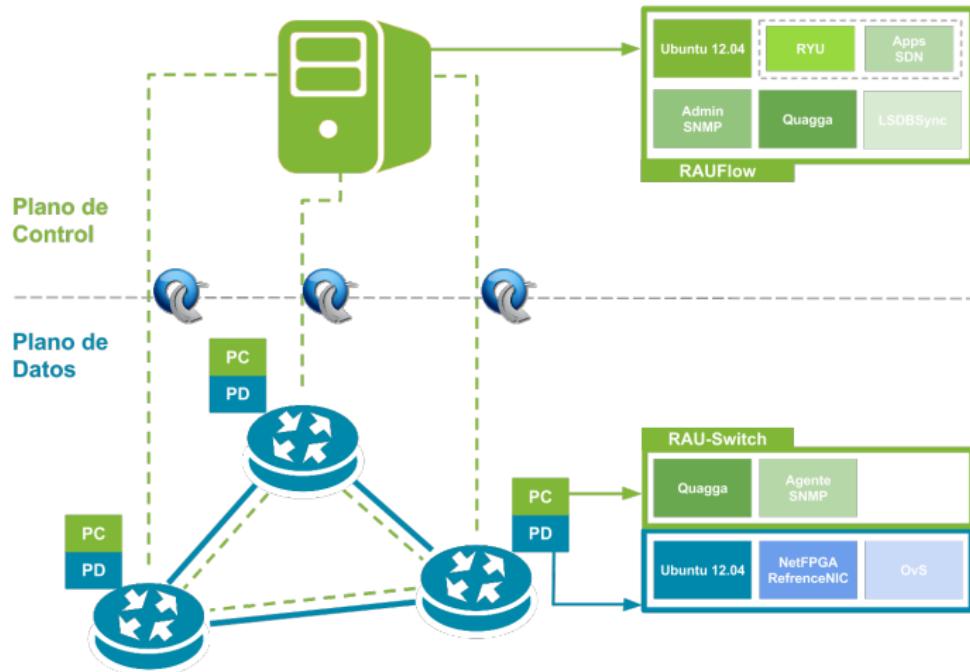
# Agente SNMP

## Agente SNMP

OpenFlow no prevé un mecanismo para informar al plano de control direccionamiento IP de un puerto en un dispositivo.

**Solución:** Utilizamos un agente SNMP en cada nodo de la red para obtener mapeo entre puertos OF y direcciones IP

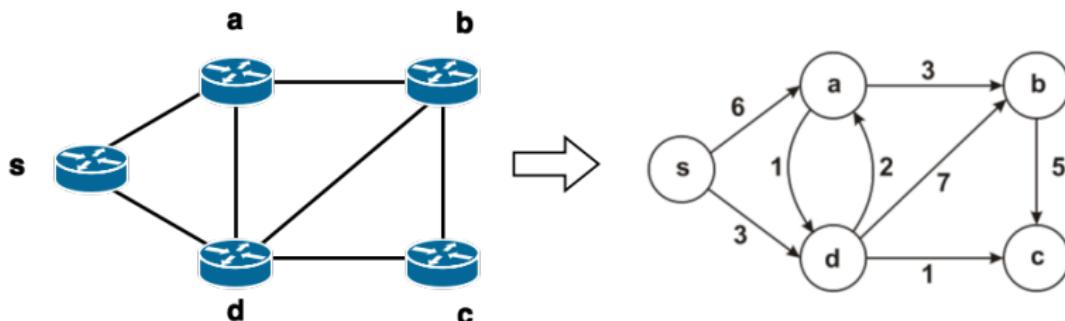
# Arquitectura Propuesta III



- 1 Introducción
- 2 Conceptos preliminares
- 3 Arquitectura propuesta
- 4 Implementación
- 5 Experimentación
- 6 Conclusiones
- 7 Trabajo a futuro

# Algoritmo de ruteo

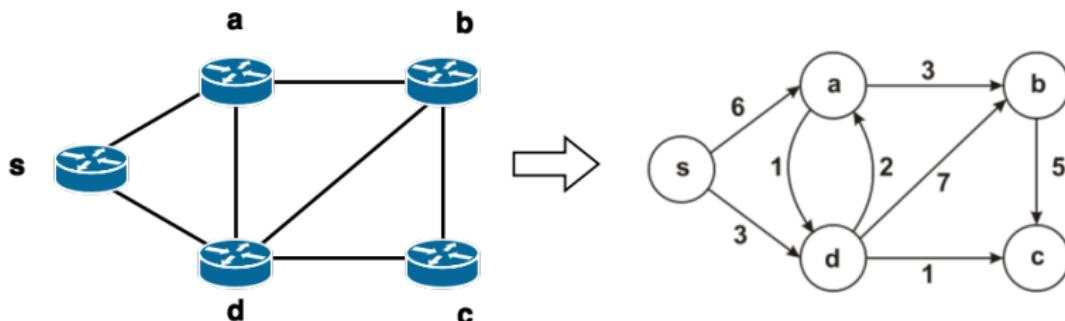
**SPF(Shortest Path First):** Algoritmo del camino más corto utilizando el algoritmo Dijkstra generalizado para multigrafos dirigidos<sup>2</sup>.



<sup>2</sup> Implementación basada en "Generalization of dijkstra's algorithm for extraction of the shortest paths in directed multigraphs".

# Algoritmo de ruteo

**SPF(Shortest Path First):** Algoritmo del camino más corto utilizando el algoritmo Dijkstra generalizado para multigrafos dirigidos<sup>2</sup>.

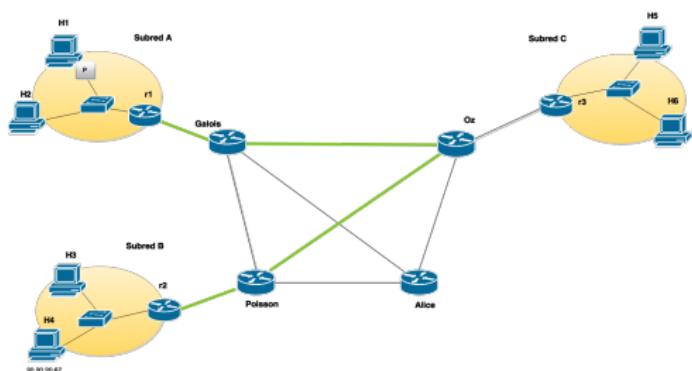
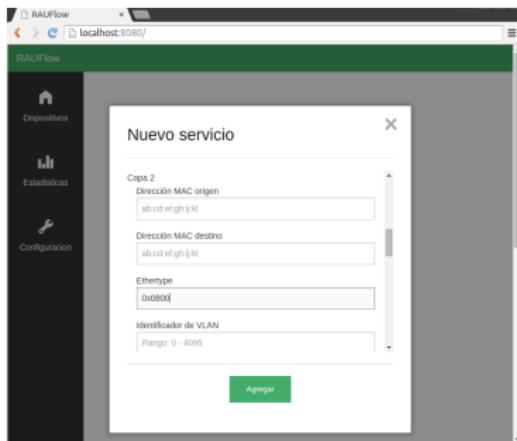


**CSPF(Constrained Shortest Path First):** Algoritmo SPF con restricciones.

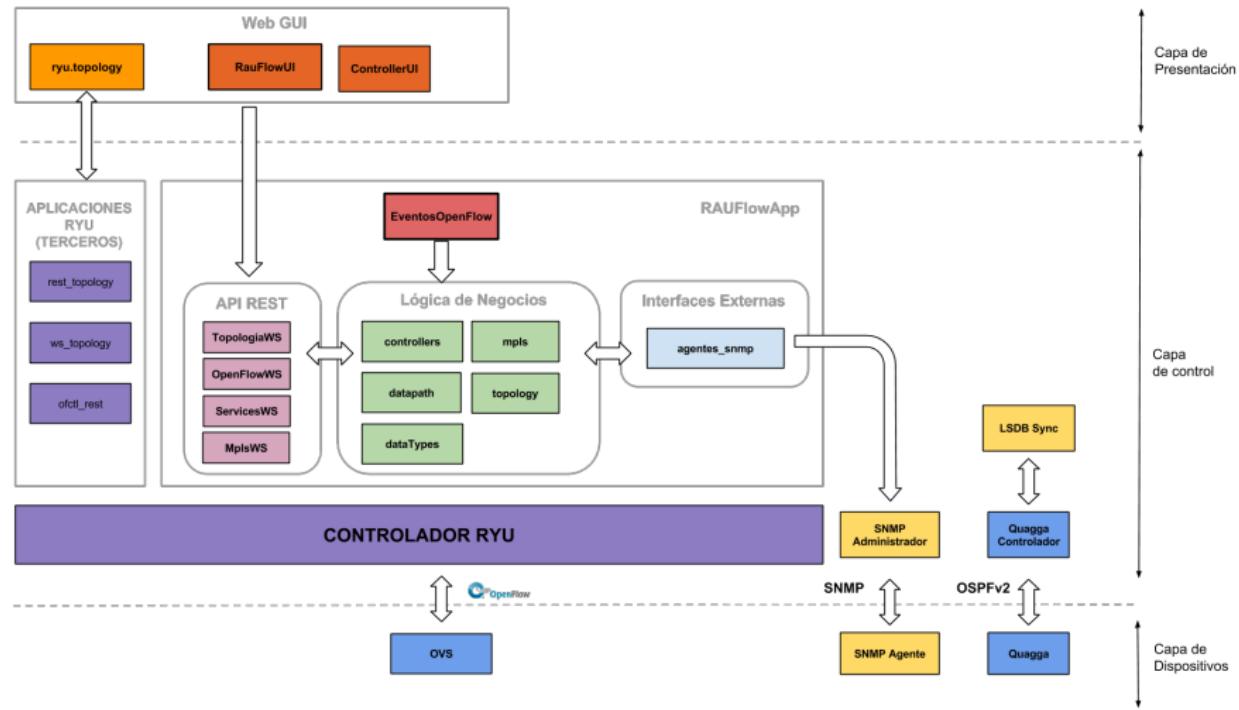
<sup>2</sup> Implementación basada en "Generalization of dijkstra's algorithm for extraction of the shortest paths in directed multigraphs".

# Clasificación de tráfico

- Se implementa clasificación de tráfico utilizando matching fields OF en los nodos de ingreso a la red
- Cada definición de servicio define una FEC que se traduce a un flujo OpenFlow para la clasificación de tráfico



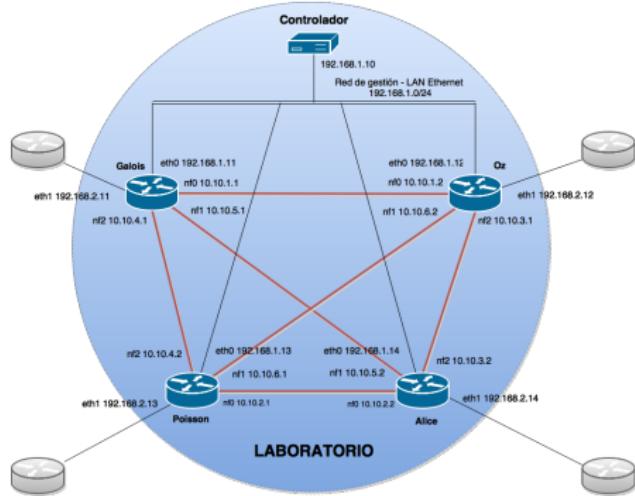
# RAUFlow



- 1 Introducción
- 2 Conceptos preliminares
- 3 Arquitectura propuesta
- 4 Implementación
- 5 Experimentación
- 6 Conclusiones
- 7 Trabajo a futuro

# Experimentación

Se construye un laboratorio de experimentación compuesto por 4 nodos RAU-Switch conectados mediante enlaces ópticos de alta velocidad en una topología full Mesh.



Se configuraron dos escenarios diferentes de VPN L3:

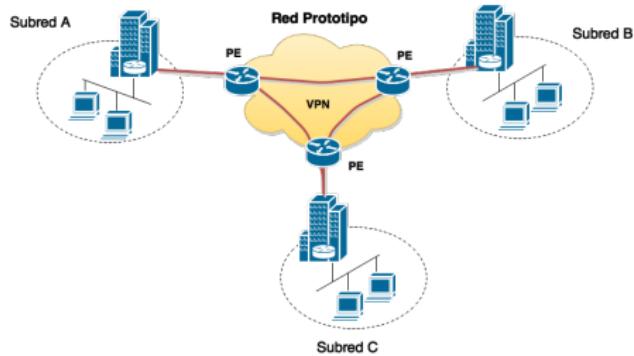
- ① E1: Una organización con tres sucursales físicamente separadas (VPN Multipunto)
- ② E2: Dos organizaciones con dos sucursales físicamente separadas c/u (VPN Punto a Punto)

Se crearon los servicios de VPN y se verificó:

- Aislamiento del espacio de direcciones IP
- Correcta clasificación de tráfico
- Definición de servicios mediante OF Matching Fields

# VPN L3 Escenario 1

Se configuró una VPN L3 multipunto conectando 3 sucursales físicamente separadas de una misma organización.

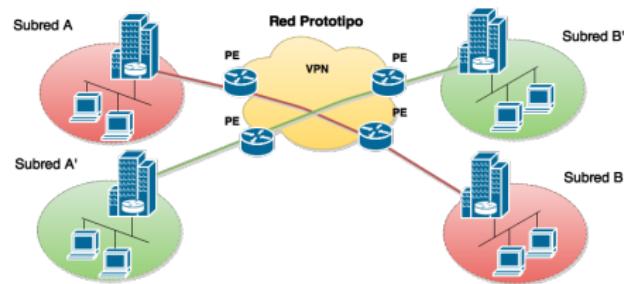


Sobre este escenario se verificó:

- Correcto enrutamiento a cada sucursal de la organización
- Correcta clasificación de tráfico
- Definición de servicios mediante OF Matching Fields

# VPN L3 Escenario 2

Se configuraron dos VPN L3 punto a punto conectando para dos organizaciones distintas y con la misma numeración IP en sus red, dos sucursales físicamente separadas.

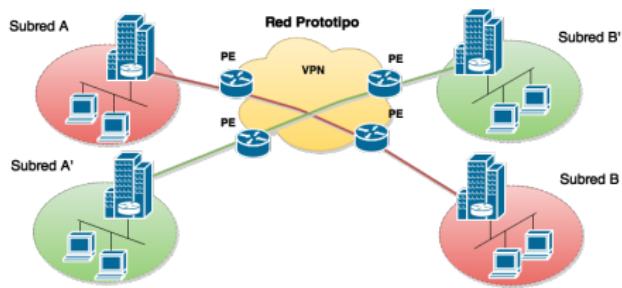


Sobre este escenario se verificó:

- Aislamiento del espacio de direcciones IP

# VPN L2

Se configuró una VPN L2 punto a punto para una organización con dos sucursales físicamente separadas.



Sobre este escenario se verificó:

- Soporte para un conjunto de 43 ethertypes diferentes independiente de la definición del servicio
- En particular los protocolos ARP, IPv4 IPv6 e IP con tags de VLAN

- 1 Introducción
- 2 Conceptos preliminares
- 3 Arquitectura propuesta
- 4 Implementación
- 5 Experimentación
- 6 Conclusiones
- 7 Trabajo a futuro

# Conclusiones I

Se alcanzaron los objetivos planteados al inicio del proyecto:

- Se exploró la aplicabilidad del enfoque SDN y el hardware NetFPGA para la construcción de un prototipo para la RAU2

# Conclusiones I

Se alcanzaron los objetivos planteados al inicio del proyecto:

- Se exploró la aplicabilidad del enfoque SDN y el hardware NetFPGA para la construcción de un prototipo para la RAU2
- Se propuso una arquitectura híbrida entre el paradigma de redes actual y el enfoque OpenFlow/SDN

# Conclusiones I

Se alcanzaron los objetivos planteados al inicio del proyecto:

- Se exploró la aplicabilidad del enfoque SDN y el hardware NetFPGA para la construcción de un prototipo para la RAU2
- Se propuso una arquitectura híbrida entre el paradigma de redes actual y el enfoque OpenFlow/SDN
- Se implementó de prototipo de aplicación de control y gestión de red utilizando SDN (RAUFlow)

# Conclusiones I

Se alcanzaron los objetivos planteados al inicio del proyecto:

- Se exploró la aplicabilidad del enfoque SDN y el hardware NetFPGA para la construcción de un prototipo para la RAU2
- Se propuso una arquitectura híbrida entre el paradigma de redes actual y el enfoque OpenFlow/SDN
- Se implementó de prototipo de aplicación de control y gestión de red utilizando SDN (RAUFlow)
- Se construyó un laboratorio de pruebas e implementó un par de casos de uso representativos para la RAU2

# Conclusiones II

Se obtuvieron resultados interesantes:

- Se confeccionó un manual para la construcción del dispositivo RAU-Switch

# Conclusiones II

Se obtuvieron resultados interesantes:

- Se confeccionó un manual para la construcción del dispositivo RAU-Switch
- Se contribuyó a la comunidad NetFPGA reportando dos BUGs

# Conclusiones II

Se obtuvieron resultados interesantes:

- Se confeccionó un manual para la construcción del dispositivo RAU-Switch
- Se contribuyó a la comunidad NetFPGA reportando dos BUGs
- Se escribió un artículo científico el cual fue aceptado en la conferencia Latin American Network Operations and Management Symposium 2015 (LANOMS)

# Conclusiones II

Se obtuvieron resultados interesantes:

- Se confeccionó un manual para la construcción del dispositivo RAU-Switch
- Se contribuyó a la comunidad NetFPGA reportando dos BUGs
- Se escribió un artículo científico el cual fue aceptado en la conferencia Latin American Network Operations and Management Symposium 2015 (LANOMS)
- Se generó un grupo de trabajo (GT SDNUY) uniendo profesionales del SeCIU, Centro de Capacitación y Desarrollo de ANTEL y del Centro Universitario de la Región Este (CURE)

- 1 Introducción
- 2 Conceptos preliminares
- 3 Arquitectura propuesta
- 4 Implementación
- 5 Experimentación
- 6 Conclusiones
- 7 Trabajo a futuro

# Trabajo a futuro

Se identifican las siguientes líneas de trabajo a futuro:

# Trabajo a futuro

Se identifican las siguientes líneas de trabajo a futuro:

- Extender el proyecto OpenFlow de la plataforma NetFPGA para soportar al menos la versión 1.3.1 del protocolo OpenFlow

# Trabajo a futuro

Se identifican las siguientes líneas de trabajo a futuro:

- Extender el proyecto OpenFlow de la plataforma NetFPGA para soportar al menos la versión 1.3.1 del protocolo OpenFlow
- Extender el algoritmo de ruteo SPF para implementar un CSPF

# Trabajo a futuro

Se identifican las siguientes líneas de trabajo a futuro:

- Extender el proyecto OpenFlow de la plataforma NetFPGA para soportar al menos la versión 1.3.1 del protocolo OpenFlow
- Extender el algoritmo de ruteo SPF para implementar un CSPF
- Incorporar en RAUFlow la capacidad de soportar múltiples caminos para un mismo Servicio, balanceo de carga, QoS e Ingeniería de tráfico

# Trabajo a futuro

Otras posibles líneas:

# Trabajo a futuro

Otras posibles líneas:

- Agregar una capa de persistencia para ciertos datos

# Trabajo a futuro

Otras posibles líneas:

- Agregar una capa de persistencia para ciertos datos
- Investigar la escalabilidad de RAUflow en topologías de red realistas

# Trabajo a futuro

Otras posibles líneas:

- Agregar una capa de persistencia para ciertos datos
- Investigar la escalabilidad de RAUflow en topologías de red realistas
- Construir una jerarquía de controladores e investigar el impacto en el rendimiento

Otras posibles líneas:

- Agregar una capa de persistencia para ciertos datos
- Investigar la escalabilidad de RAUflow en topologías de red realistas
- Construir una jerarquía de controladores e investigar el impacto en el rendimiento
- Incorporar más dimensiones a la definición de un servicio (tiempo)

# ¿Preguntas?