

# **Proyecto de grado**

## **Escalabilidad de Redes Definidas por Software en la Red Académica**

**Santiago Vidal**

Tutores:

Dr. Eduardo Grampín

MSc. Martín Giachino

Instituto de Computación

Facultad de Ingeniería

Universidad de la República

5 de octubre de 2016

## **Introducción**

## **Conceptos previos & RAUFlow**

## **Entorno virtual**

## **Pruebas de escala**

## **Conclusiones**

# Introducción

## Conceptos previos & RAUFlow

## Entorno virtual

## Pruebas de escala

## Conclusiones

# Introducción

## Red Académica Uruguaya (RAU)

- ▶ Emprendimiento de la Universidad de la República, administrado por el Servicio Central de Informática Universitario (SeCIU).
- ▶ Red que conecta instituciones académicas, centros de investigación e instituciones gubernamentales.
- ▶ Parte de la RedClara.

## RAU2

RAU2 es un proyecto para reemplazar la infraestructura actual, con el objetivo de brindar más y mejores servicios a las instituciones.

# Proyecto RRAP

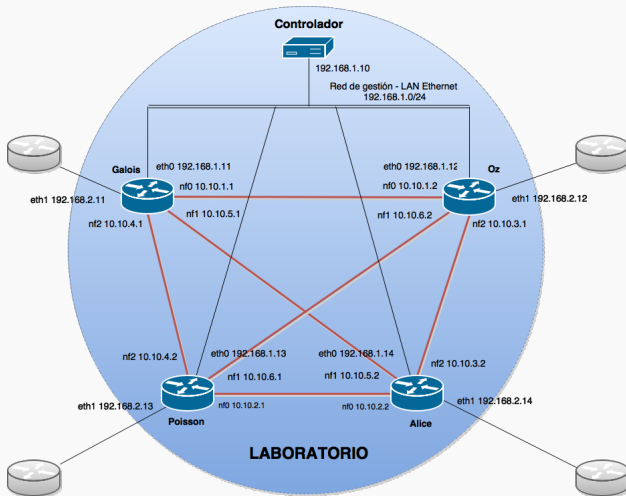
## Routers Reconfigurables de Altas Prestaciones

(Emiliano Viotti, Rodrigo Amaro):

- ▶ Proyecto de grado que terminó en agosto de 2015.
- ▶ Construyó un prototipo para la RAU2, basado en SDN.
- ▶ Desarrolló una aplicación para gestión de redes llamada **RAUFlow**, que implementa clasificación y separación de tráfico.

# Proyecto RRAP

Prototipo físico para pruebas funcionales:



# Motivación

Contestar las siguientes preguntas:

# Motivación

Contestar las siguientes preguntas:

1. ¿Cómo podemos seguir trabajando sobre la arquitectura RAUflow sin ser limitados por el prototipo físico?



# Motivación

Contestar las siguientes preguntas:

1. ¿Cómo podemos seguir trabajando sobre la arquitectura RAUflow sin ser limitados por el prototipo físico?
2. ¿RAUFlow funciona con topologías más grandes?

# Motivación

Contestar las siguientes preguntas:

1. ¿Cómo podemos seguir trabajando sobre la arquitectura RAUflow sin ser limitados por el prototipo físico?
2. ¿RAUFlow funciona con topologías más grandes?
3. ¿Tiene buena escalabilidad?

# Resultados esperados

# Resultados esperados

1. Estado del arte en las aplicaciones de SDN (con foco en VPNs), y las herramientas de virtualización disponibles.

# Resultados esperados

1. Estado del arte en las aplicaciones de SDN (con foco en VPNs), y las herramientas de virtualización disponibles.
2. Una herramienta que permita virtualizar la arquitectura RAUFlow para pruebas y desarrollo.

# Resultados esperados

1. Estado del arte en las aplicaciones de SDN (con foco en VPNs), y las herramientas de virtualización disponibles.
2. Una herramienta que permita virtualizar la arquitectura RAUFlow para pruebas y desarrollo.
3. Diseño e implementación de pruebas para estudiar la escalabilidad de RAUFlow.

## Introducción

## Conceptos previos & RAUFlow

## Entorno virtual

## Pruebas de escala

## Conclusiones

# Diapo1



Introducción

Conceptos previos & RAUFlow

**Entorno virtual**

Pruebas de escala

Conclusiones

# Objetivo

Poder utilizar la arquitectura RAUFlow y RAUSwitch en un entorno virtual para:

- ▶ Experimentos y pruebas.
- ▶ Desarrollo de nuevas funcionalidades sobre RAUFlow.
- ▶ Investigación sobre esquemas híbridos en general.

# Requerimientos

Requerimientos funcionales:

- 1.** RAUSwitch virtuales:
  - 1.1** OpenFlow 1.3
  - 1.2** OSPF
  - 1.3** SNMP (no esencial)
- 2.** Hosts virtuales
- 3.** Controlador RAUFlow

# Requerimientos

Requerimientos funcionales:

- 1.** RAUSwitch virtuales:
  - 1.1** OpenFlow 1.3
  - 1.2** OSPF
  - 1.3** SNMP (no esencial)
- 2.** Hosts virtuales
- 3.** Controlador RAUFlow

Requerimientos no funcionales:

- 1.** Configurabilidad / Usabilidad
- 2.** Escalabilidad

## Siguiente paso

Se descarta una construcción desde cero



Hay que encontrar una herramienta que cumpla los requerimientos

# Elección de una herramienta

## Herramientas orientadas a SDN

- ▶ Algunas no soportan OpenFlow 1.3
- ▶ Algunas no permiten un controlador externo.
- ▶ **Ninguna contempla switches híbridos!**

# Elección de una herramienta

## Herramientas orientadas a SDN

- ▶ Algunas no soportan OpenFlow 1.3
- ▶ Algunas no permiten un controlador externo.
- ▶ **Ninguna contempla switches híbridos!**

## Herramientas de propósito general

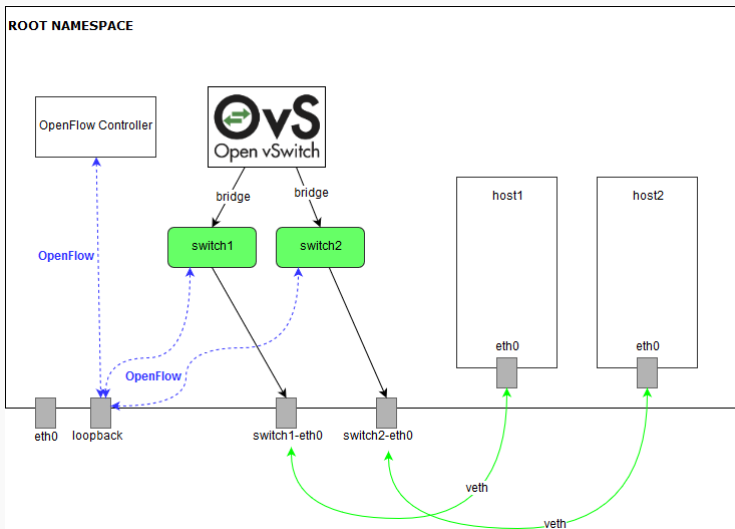
- ▶ Algunas no tienen buena configurabilidad.
- ▶ La **escalabilidad** es un gran problema.

# Mininet

- ▶ Emulador de redes.
- ▶ Comúnmente utilizado para experimentar con SDN y OpenFlow.
- ▶ Ofrece Hosts y Switches.
- ▶ Virtualización ligera (containers).
- ▶ Cumple todos los requerimientos **excepto** el soporte para switches híbridos.
- ▶ Pero permite al usuario definir sus propias clases de nodos para extender las funcionalidades de las clases que vienen por defecto.



# Arquitectura de Mininet



# Problema con Mininet tradicional

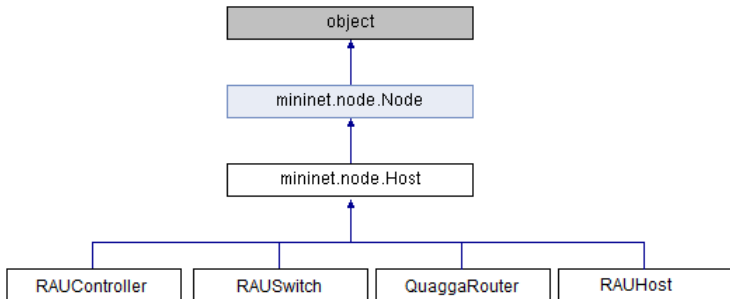
- ▶ Los switches están en el root namespace, así que no es posible que cada uno ejecute su instancia de Quagga.
- ▶ No es posible poner a cada Switch en su propio namespace ya que Open vSwitch no tendría acceso a ellos.
- ▶ Si los switches están en su propio namespace, el controlador OpenFlow (RAUFlow) no puede comunicarse con ellos a través de la interfaz de loopback.

## Problema con Mininet tradicional

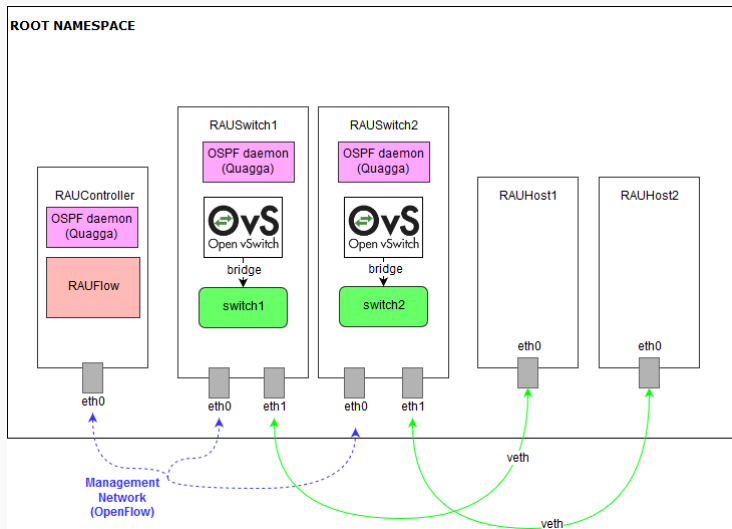
- ▶ Los switches están en el root namespace, así que no es posible que cada uno ejecute su instancia de Quagga.
- ▶ No es posible poner a cada Switch en su propio namespace ya que Open vSwitch no tendría acceso a ellos.
- ▶ Si los switches están en su propio namespace, el controlador OpenFlow (RAUFlow) no puede comunicarse con ellos a través de la interfaz de loopback.

**Solución:** utilizar Mininet pero como emulador de propósito general.

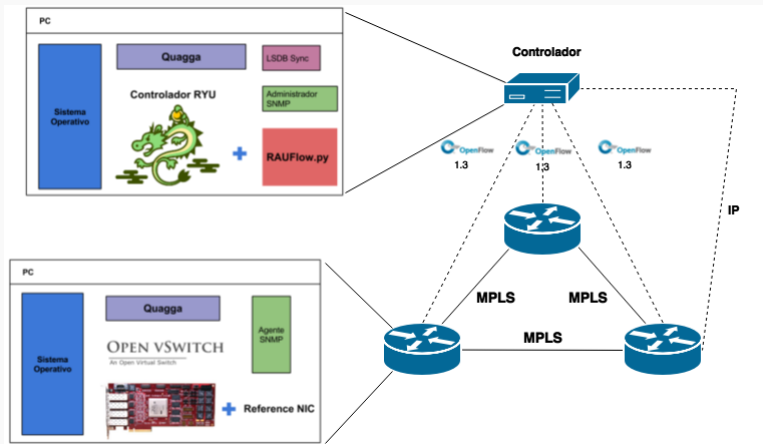
# Diseño de la solución



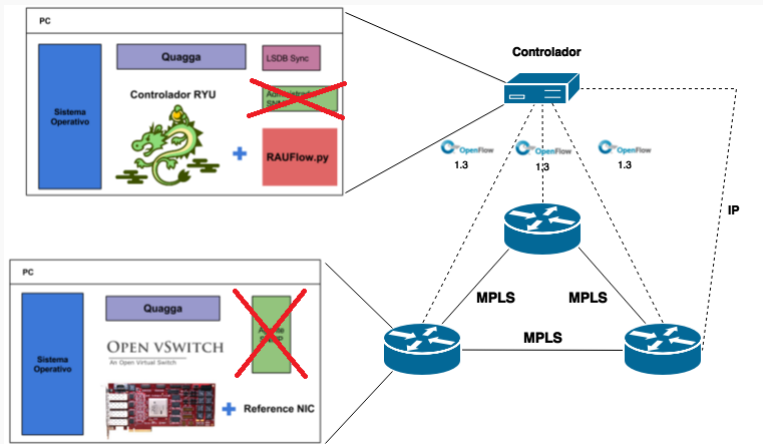
# Arquitectura del entorno construido



# Eliminación de SNMP



# Eliminación de SNMP



# Eliminación de SNMP

El envío de datos de las interfaces pasa a implementarse con Open vSwitch (por fuera de OpenFlow).

## Ventajas

- ▶ Reduce complejidad de la arquitectura.
- ▶ Reduce carga de cómputo en los switches.



# Verificación funcional

Con el entorno construido, el siguiente paso es probar distintos escenarios y topologías para detectar:

- ▶ Problemas con el entorno virtual.
- ▶ Problemas con la arquitectura/código de RAUFlow.

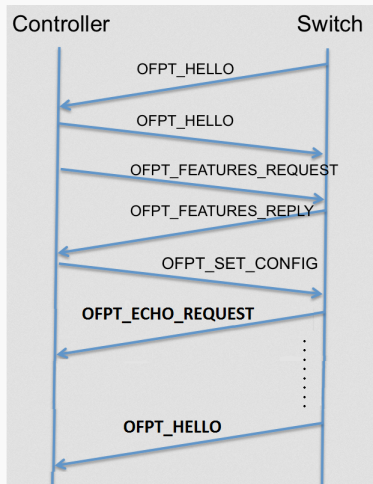
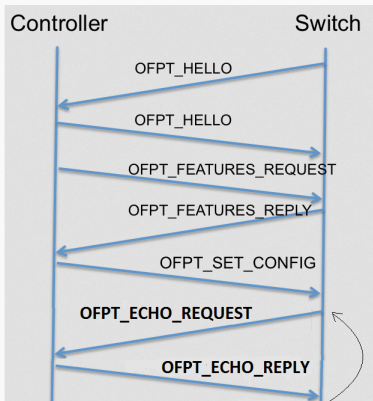
# Problemas encontrados

1. **Error en el código de RAUFlow:** error en el algoritmo del camino óptimo. Provocaba una excepción de Python.
2. **Error en el código de RAUFlow:** error en el código que instala los flujos OpenFlow en los nodos. Provocaba que los flujos en cada nodo de un camino tuvieran incorrecto puerto de entrada.

# Problemas encontrados

3. **Posible problema** en el módulo LSDB Sync para leer base de datos topológica de OSPF cuando la topología es muy grande (librería Telnetlib de Python).
4. **Posible problema** de comunicación en la red de gestión cuando hay muchos switches.

# Problemas encontrados



## Introducción

## Conceptos previos & RAUFlow

## Entorno virtual

## Pruebas de escala

## Conclusiones

# Diapo1

## Introducción

## Conceptos previos & RAUFlow

## Entorno virtual

## Pruebas de escala

## Conclusiones

# Diapo1