

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

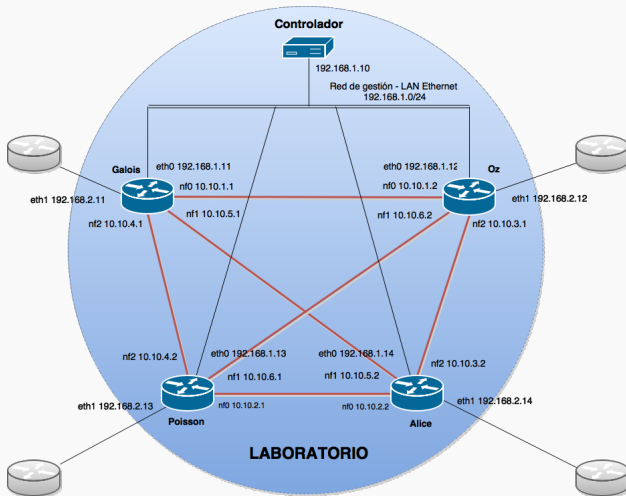
Routers Reconfigurables de Altas Prestaciones

(Emiliano Viotti, Rodrigo Amaro):

- ▶ Proyecto de grado que terminó en agosto de 2015.
- ▶ Desarrolló una arquitectura de red basada en SDN llamada **RAUFlow**, que implementa Redes Privadas Virtuales (VPN).
- ▶ Construyó un prototipo físico para validación funcional.

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. Una **herramienta** que permita virtualizar la arquitectura RAUFlow para pruebas y desarrollo.

Resultados esperados

1. Una **herramienta** que permita virtualizar la arquitectura RAUFlow para pruebas y desarrollo.
2. Diseño e implementación de **pruebas** para estudiar la escalabilidad de RAUFlow.

Introducción

Conceptos previos & RAUFlow

Entorno virtual

Pruebas de escala

Conclusiones

Conceptos previos

Estudiemos algunos conceptos previos relacionados a RAUFlow:

1. VPN
2. MPLS
3. Software Defined Networking
4. OpenFlow

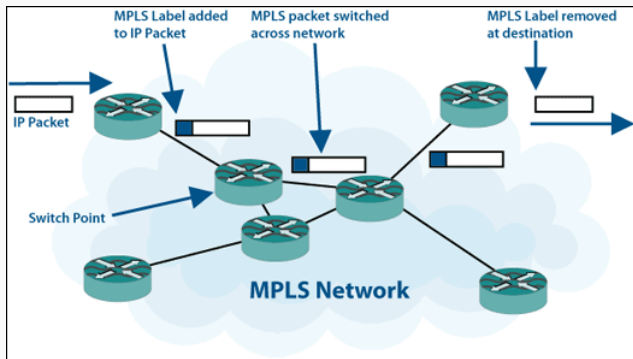
VPN

Red privada que se extiende a través de una red pública, como Internet. Puede ser de capa 2 (enlace) o 3 (red) del modelo OSI.



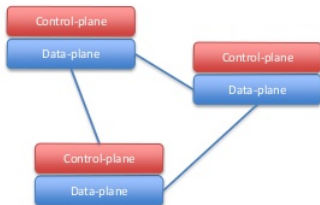
MPLS

Mecanismo de transporte de datos basado en la conmutación de etiquetas. Es la solución de facto para la implementación de servicios de VPN.

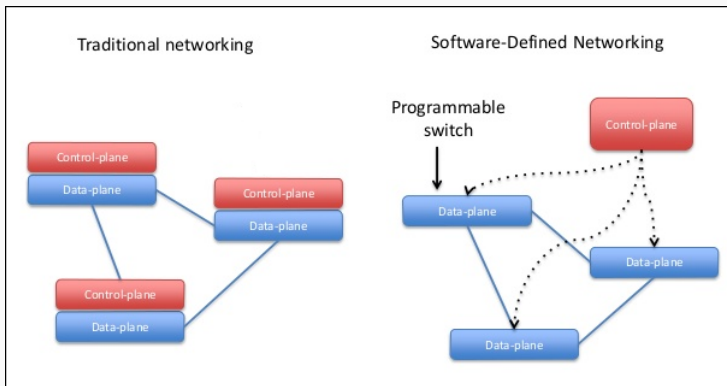


Redes Definidas por Software (SDN)

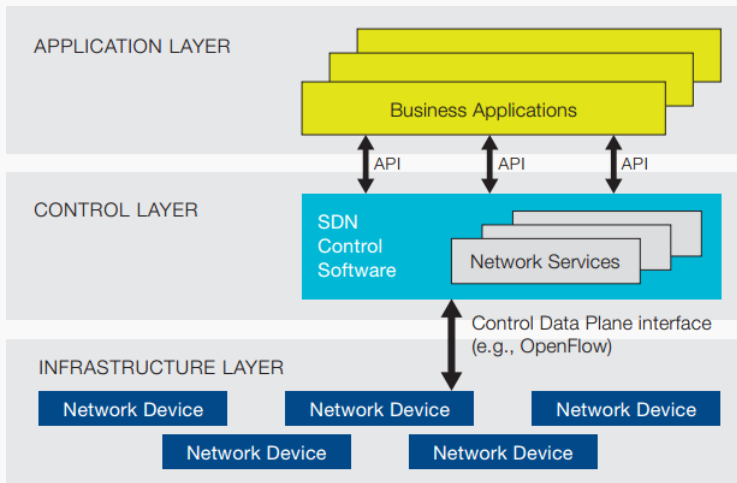
Traditional networking



Redes Definidas por Software (SDN)



Redes Definidas por Software (SDN)



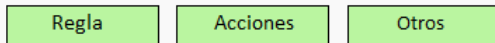
OpenFlow

- ▶ Provee una forma de abstraer las capacidades de un dispositivo.
- ▶ Protocolo de comunicación con el controlador OpenFlow.

OpenFlow

- ▶ Provee una forma de abstraer las capacidades de un dispositivo.
- ▶ Protocolo de comunicación con el controlador OpenFlow.

Flujo OpenFlow:

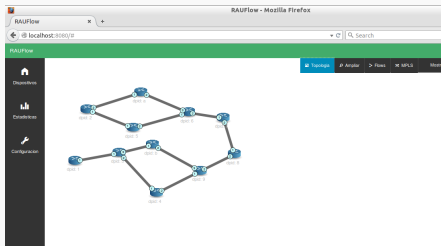


- ▶ Regla: Define el tipo de tráfico.
- ▶ Acciones: Define qué se debe hacer con los paquetes del flujo (Drop, Output, Agregar/Quitar etiqueta MPLS)
- ▶ Otros: Estadísticas, Prioridad, Timeout, etc.

RAUFlow

Aplicación para control de red basada en el controlador Ryu.

1. Implementa servicios de VPN de capa 2 y 3.
2. OpenFlow y MPLS.
3. OSPF y SNMP.

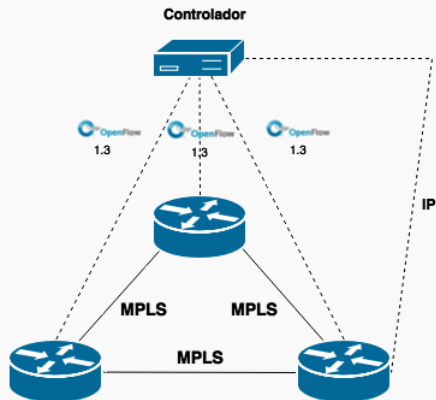


The screenshot shows the RAUFlow web interface with the 'Nuevo servicio' (New service) configuration form open. The form is titled 'Cape 2' and contains the following fields:

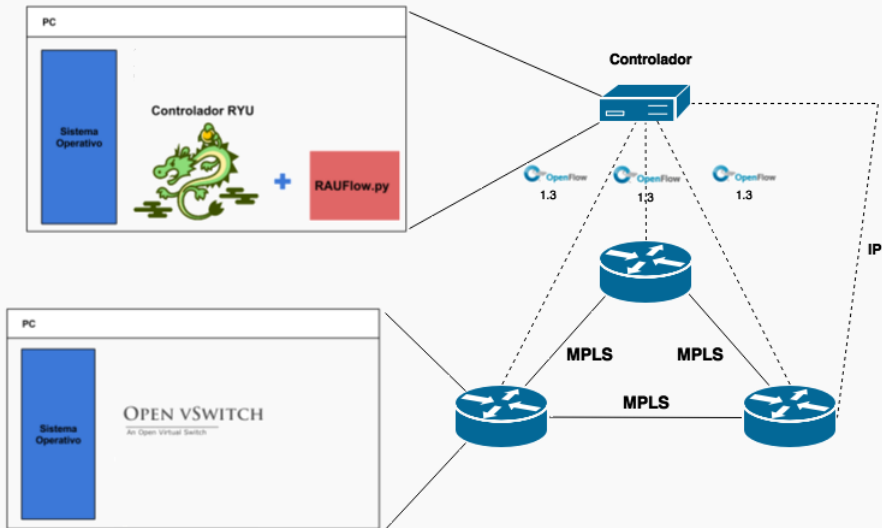
- Dirección MAC origen:**
- Dirección MAC destino:**
- Ethertype:**
- Identificador de VLAN:**

At the bottom of the form is a green button labeled 'Agregar'.

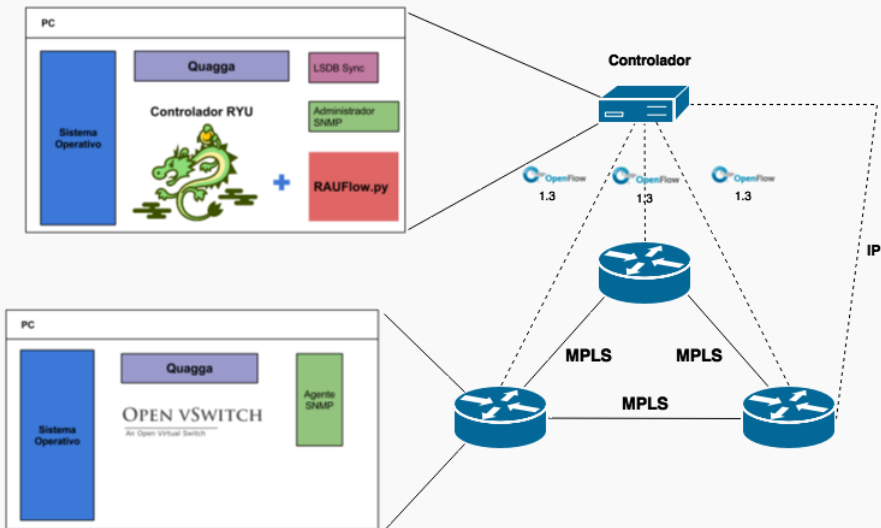
RAUFlow



RAUFlow



RAUFlow



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!**

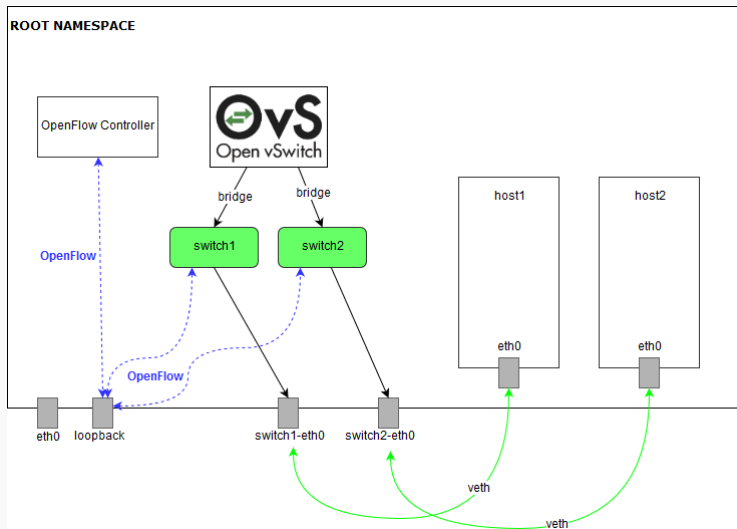
Herramientas de propósito general

- ▶ 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.
- ▶ 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

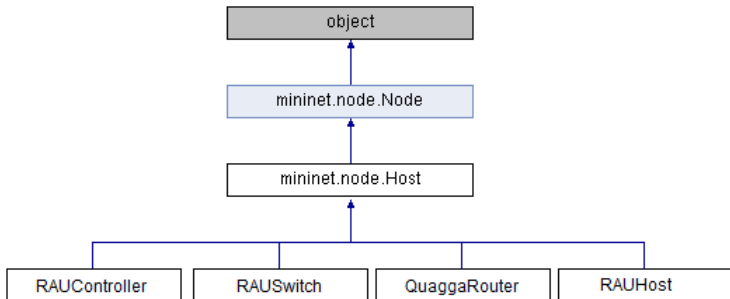
- ▶ Los switches están en el root namespace, así que no es posible que cada uno ejecute su instancia de Quagga.
- ▶ No es posible poner a cada Switch en su propio namespace ya que Open vSwitch no tendría acceso a ellos.
- ▶ Si los switches están en su propio namespace, el controlador OpenFlow (RAUFlow)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

Problema con Mininet tradicional

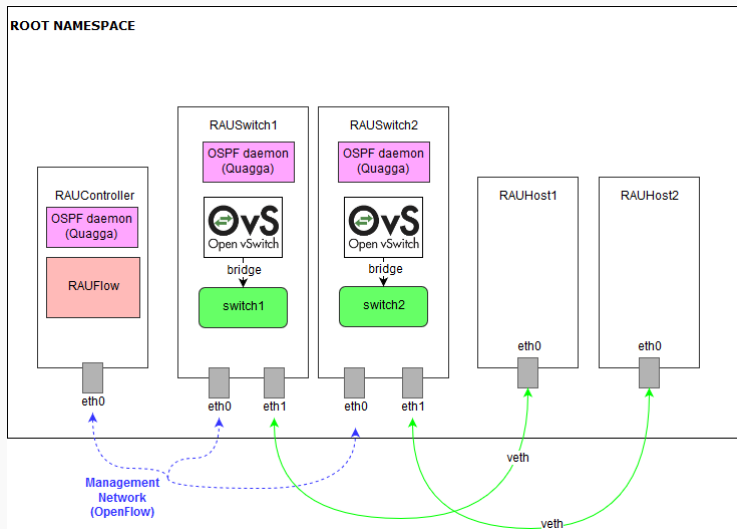
- ▶ Los switches están en el root namespace, así que no es posible que cada uno ejecute su instancia de Quagga.
- ▶ No es posible poner a cada Switch en su propio namespace ya que Open vSwitch no tendría acceso a ellos.
- ▶ Si los switches están en su propio namespace, el controlador OpenFlow (RAUFlow)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

Solución: utilizar Mininet pero como emulador de propósito general.

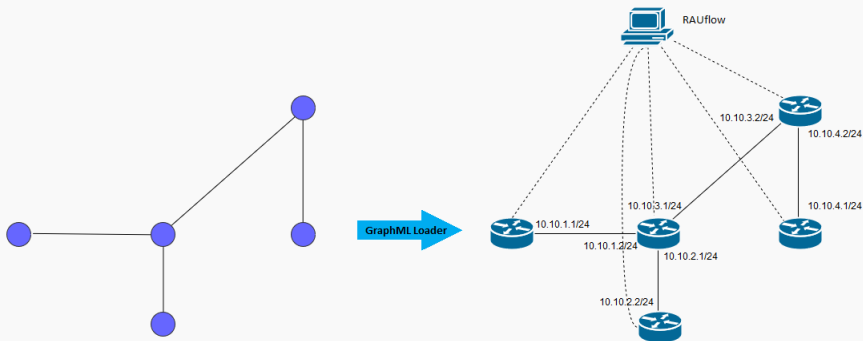
Diseño de la solución



Arquitectura del entorno construido



GraphML Loader



Verificación funcional

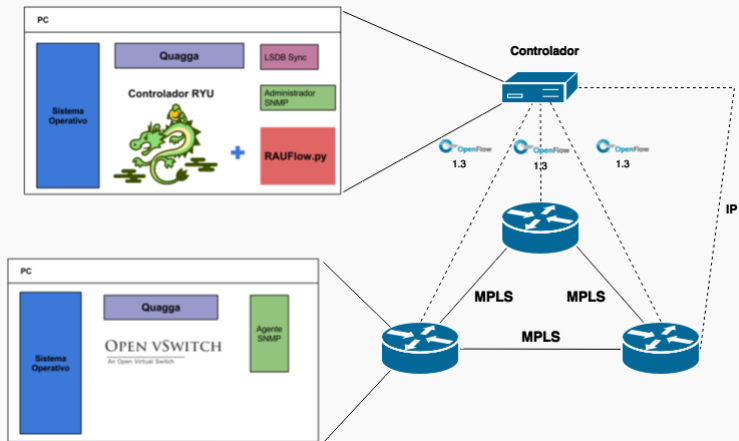
Con el entorno construido, el siguiente paso es probar distintos escenarios y topologías para detectar:

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

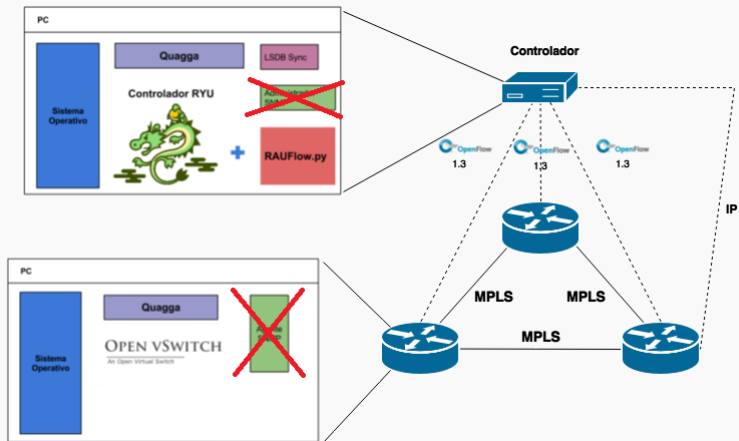
Problemas encontrados

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.

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.

Introducción

Conceptos previos & RAUFlow

Entorno virtual

Pruebas de escala

Conclusiones

Pruebas de escala

Podemos analizar la escalabilidad desde dos frentes:

- 1. Variable:** Tamaño de la topología
Analizar: Creación de los servicios
- 2. Variable:** Nivel de carga
Analizar: Rendimiento de la red

Pruebas de escala

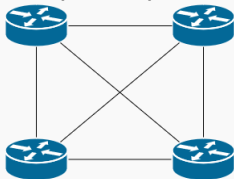
Podemos analizar la escalabilidad desde dos frentes:

- 1. Variable:** Tamaño de la topología
Analizar: Creación de los servicios
- 2. Variable:** Nivel de carga
Analizar: Rendimiento de la red

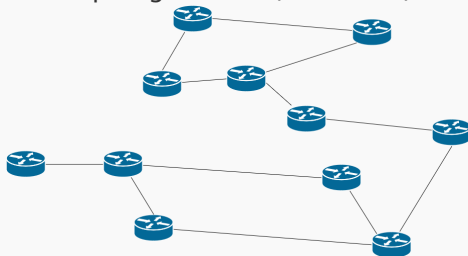
Observación: las pruebas no nos dirán valores reales, pero sí nos permiten detectar comportamientos y tendencias.

Pruebas de escala: topologías

Topología básica
(prototipo)

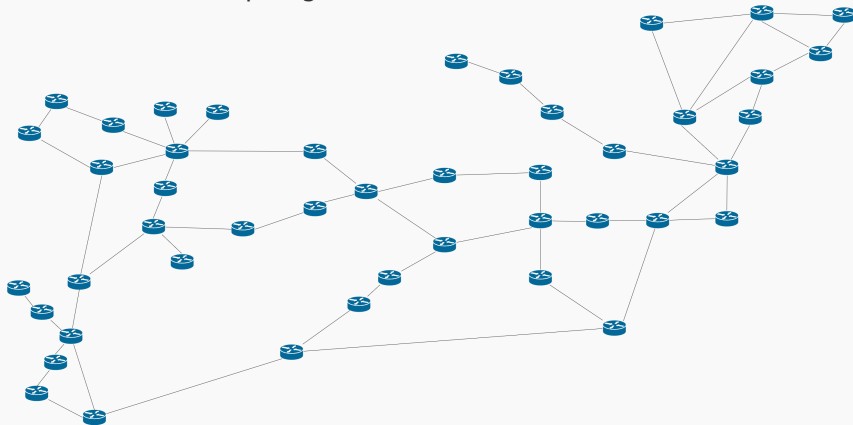


Topología chica (11 nodos)



Pruebas de escala: topologías

Topología mediana (45 nodos)



Tiempos de creación de VPN

VPN de capa 2 (en ms)

Largo del camino	Básica	Chica	Mediana	Grande
1	191	234	311	592
2	N/C	296	611	1204
4	N/C	299	453	1826
6	N/C	313	491	1662
8	N/C	324	567	2692
10	N/C	N/C	481	755
12	N/C	N/C	536	2480

VPN de capa 3 (en ms)

Largo del camino	Básica	Chica	Mediana	Grande
1	17	22	103	418
2	N/C	28	119	675
4	N/C	44	129	1237
6	N/C	53	158	696
8	N/C	60	196	1099
10	N/C	N/C	196	555
12	N/C	N/C	193	971

Tiempos de creación de VPN

VPN de capa 2 (en ms)

Largo del camino	Básica	Chica	Mediana	Grande
1	191	234	311	592
2	N/C	296	611	1204
4	N/C	299	453	1826
6	N/C	313	491	1662
8	N/C	324	567	2692
10	N/C	N/C	481	755
12	N/C	N/C	536	2480




- tiempo

VPN de capa 3 (en ms)


Largo del camino	Básica	Chica	Mediana	Grande
1	17	22	103	418
2	N/C	28	119	675
4	N/C	44	129	1237
6	N/C	53	158	696
8	N/C	60	196	1099
10	N/C	N/C	196	555
12	N/C	N/C	193	971

Tiempos de creación de VPN



+ tiempo

Largo del camino	Básica	Chica	Mediana	Grande
1	191	234	311	592
2	N/C	296	611	1204
4	N/C	299	453	1826
6	N/C	313	491	1662
8	N/C	324	567	2692
10	N/C	N/C	481	755
12	N/C	N/C	536	2480



+ tiempo

Largo del camino	Básica	Chica	Mediana	Grande
1	17	22	103	418
2	N/C	28	119	675
4	N/C	44	129	1237
6	N/C	53	158	696
8	N/C	60	196	1099
10	N/C	N/C	196	555
12	N/C	N/C	193	971

Tiempos de creación de VPN

VPN de capa 2 (en ms)

Largo del camino	Básica	Chica	Mediana	Grande
1	191	234	311	592
2	N/C	296	611	1204
4	N/C	299	453	1826
6	N/C	313	491	1662
8	N/C	324	567	2692
10	N/C	N/C	481	755
12	N/C	N/C	536	2480



VPN de capa 3 (en ms)

Largo del camino	Básica	Chica	Mediana	Grande
1	17	22	103	418
2	N/C	28	119	675
4	N/C	44	129	1237
6	N/C	53	158	696
8	N/C	60	196	1099
10	N/C	N/C	196	555
12	N/C	N/C	193	971

Tiempos de creación de VPN

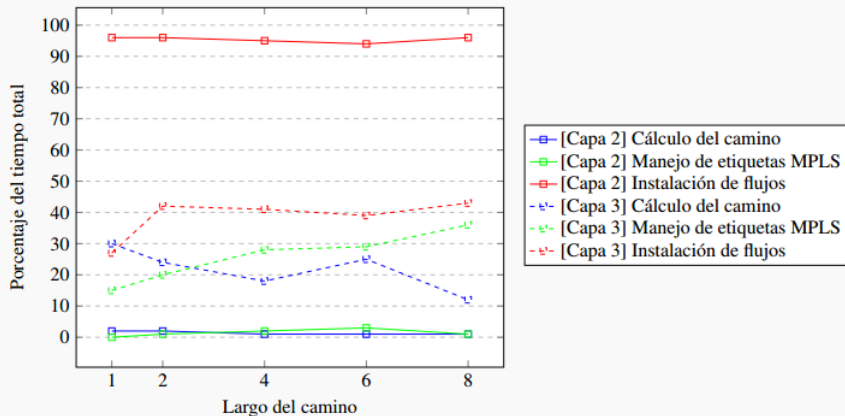
Hagamos un análisis más fino.

Agregando **timestamps** al código podemos ver cuánto tiempo dedica RAUflow a cada tarea:

- ▶ Cálculo del camino óptimo
- ▶ Manejo de etiquetas MPLS
- ▶ Instalación de flujos OpenFlow

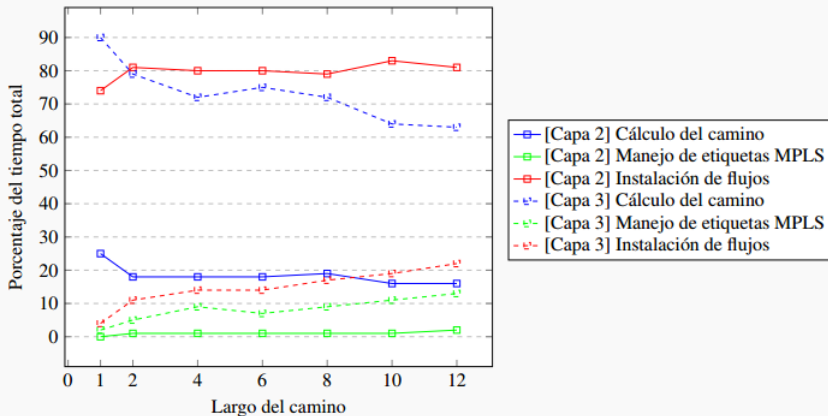
Descomposición del tiempo de creación

Topología chica:



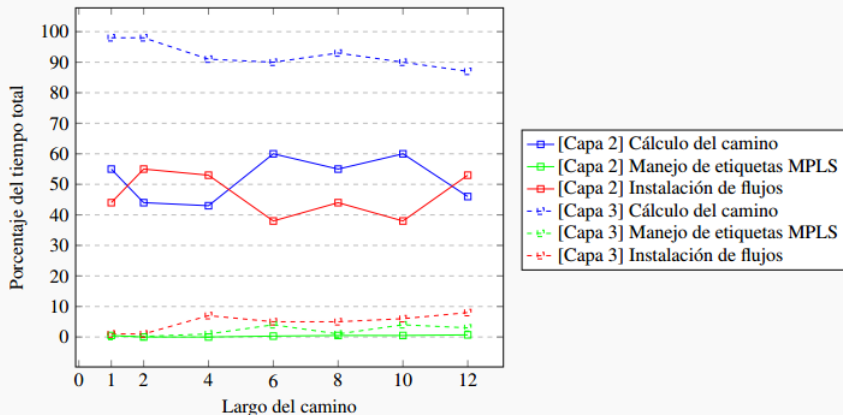
Descomposición del tiempo de creación

Topología mediana:



Descomposición del tiempo de creación

Topología grande:



Pruebas de escala: servicios

Podemos analizar la escalabilidad desde dos frentes:

- 1. Variable:** Tamaño de la topología
Analizar: Creación de los servicios
- 2. Variable:** Nivel de carga
Analizar: Rendimiento de la red

Pruebas de escala: servicios

Podemos analizar la escalabilidad desde dos frentes:

- 1. Variable:** Tamaño de la topología
Analizar: Creación de los servicios
- 2. Variable:** Nivel de carga
Analizar: Rendimiento de la red

Lo ideal sería tener **modelos de tráfico**, pero no están disponibles.

Pruebas de escala: servicios

Podemos analizar la escalabilidad desde dos frentes:

- 1. Variable:** Tamaño de la topología
Analizar: Creación de los servicios
- 2. Variable:** Nivel de carga
Analizar: Rendimiento de la red

Lo ideal sería tener **modelos de tráfico**, pero no están disponibles.

Pero sí podemos crear **muchos servicios** y ver que efecto genera.

Cantidad de servicios - throughput

Primer enfoque: determinar si la cantidad de servicios afecta a los RAUSwitch de forma individual (flujos).

Cantidad de servicios - throughput

Primer enfoque: determinar si la cantidad de servicios afecta a los RAUSwitch de forma individual (flujos).

# de VPNs	Throughput (Kbits/s)
1	1287
3000	1286
6000	1265
9000	1270
12000	1290
15000	1280

Cantidad de servicios - throughput

Primer enfoque: determinar si la cantidad de servicios afecta a los RAUSwitch de forma individual (flujos).

# de VPNs	Throughput (Kbits/s)
1	1287
3000	1286
6000	1265
9000	1270
12000	1290
15000	1280

15.000 VPN = 1.260.000 flujos

Cantidad de servicios - throughput

Primer enfoque: determinar si la cantidad de servicios afecta a los RAUSwitch de forma individual (flujos).

# de VPNs	Throughput (Kbits/s)
1	1287
3000	1286
6000	1265
9000	1270
12000	1290
15000	1280

**Open vSwitch hace
cacheo de flujos!**



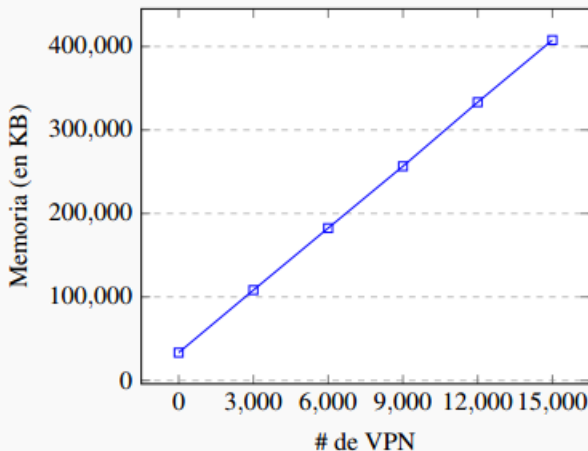
15.000 VPN = 1.260.000 flujos

Cantidad de servicios - Memoria

Segundo enfoque: consumo de memoria de RAUFlow.

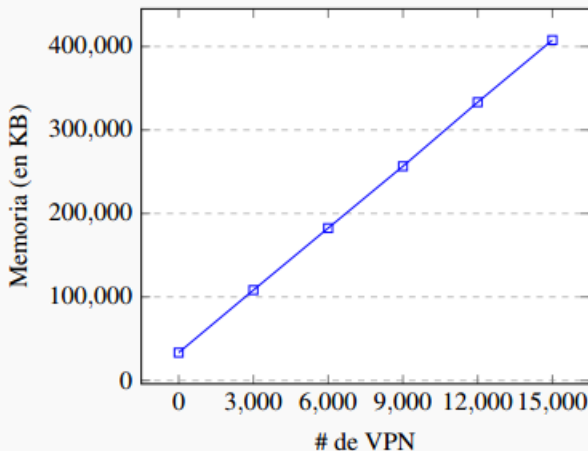
Cantidad de servicios - Memoria

Segundo enfoque: consumo de memoria de RAUFlow.



Cantidad de servicios - Memoria

Segundo enfoque: consumo de memoria de RAUFlow.



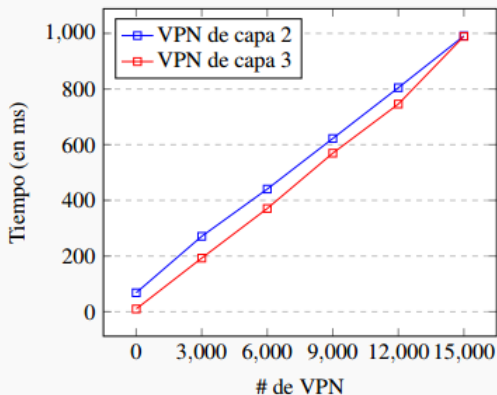
No es un consumo ineficiente, pero es deseable que se implemente persistencia.

Cantidad de servicios - nuevo servicio

Efecto inesperado: efecto sobre la creación de nuevos servicios.

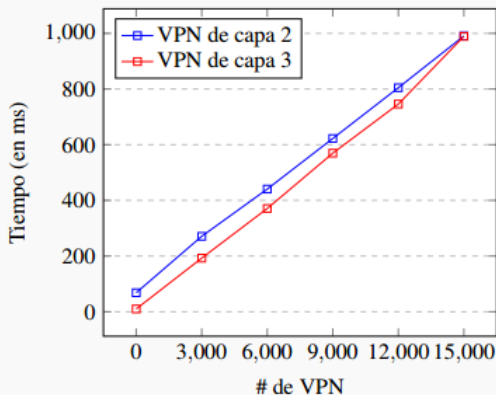
Cantidad de servicios - nuevo servicio

Efecto inesperado: efecto sobre la creación de nuevos servicios.



Cantidad de servicios - nuevo servicio

Efecto inesperado: efecto sobre la creación de nuevos servicios.



Falta determinar porqué ocurre (probablemente esté relacionado al consumo de memoria).

Introducción

Conceptos previos & RAUFlow

Entorno virtual

Pruebas de escala

Conclusiones

Conclusiones

- ▶ Se construyó una herramienta capaz de emular arquitecturas híbridas legacy/SDN.

Conclusiones

- ▶ Se construyó una herramienta capaz de emular arquitecturas híbridas legacy/SDN.
- ▶ Se corrigieron errores de implementación en RAUFlow, y se detectaron posibles problemas que podrían afectar un despliegue real.

Conclusiones

- ▶ Se construyó una herramienta capaz de emular arquitecturas híbridas legacy/SDN.
- ▶ Se corrigieron errores de implementación en RAUFlow, y se detectaron posibles problemas que podrían afectar un despliegue real.
- ▶ Se hizo un cambio en la arquitectura de RAUFlow (eliminación de SNMP), que reduce su complejidad y aumenta su rendimiento.

Conclusiones

- ▶ Se construyó una herramienta capaz de emular arquitecturas híbridas legacy/SDN.
- ▶ Se corrigieron errores de implementación en RAUFlow, y se detectaron posibles problemas que podrían afectar un despliegue real.
- ▶ Se hizo un cambio en la arquitectura de RAUFlow (eliminación de SNMP), que reduce su complejidad y aumenta su rendimiento.
- ▶ Se hizo un análisis de escalabilidad sobre RAUFlow, estudiando su comportamiento ante dos variables: topologías y servicios.

Conclusiones (2)

- ▶ Se contribuyó a un artículo científico que fue aceptado en la conferencia ACM SIGCOMM Workshop on Fostering Latin American Research in Data Communication Networks (LANCOMM 2016).
- ▶ El artículo también fue aceptado en formato de póster en la conferencia Spring School on Networks (SSN 2016) a llevarse a cabo en Chile.

Trabajo futuro

Sobre el emulador y las pruebas:

- ▶ Agregarle una interfaz gráfica para que sea más fácil de usar (MiniEdit).

Trabajo futuro

Sobre el emulador y las pruebas:

- ▶ Agregarle una interfaz gráfica para que sea más fácil de usar (MiniEdit).
- ▶ Explorar a fondo algunos comportamientos detectados (por ejemplo, el efecto acumulativo en la creación de VPNs).

Trabajo futuro

Sobre el emulador y las pruebas:

- ▶ Agregarle una interfaz gráfica para que sea más fácil de usar (MiniEdit).
- ▶ Explorar a fondo algunos comportamientos detectados (por ejemplo, el efecto acumulativo en la creación de VPNs).
- ▶ Hacer pruebas más realistas, con modelos de tráfico que reflejen la actividad de la RAU.

Trabajo futuro

Sobre RAUFlow:

- ▶ Mejorar la implementación de VPN (diferentes caminos, load balacing, ingeniería de tráfico)

Trabajo futuro

Sobre RAUFlow:

- ▶ Mejorar la implementación de VPN (diferentes caminos, load balacing, ingeniería de tráfico)
- ▶ Implementar nuevas funcionalidades de acuerdo a los requerimientos de la RAU.

Trabajo futuro

Sobre RAUFlow:

- ▶ Mejorar la implementación de VPN (diferentes caminos, load balacing, ingeniería de tráfico)
- ▶ Implementar nuevas funcionalidades de acuerdo a los requerimientos de la RAU.
- ▶ Implementar persistencia, para que no tenga todos los datos en memoria.

¿Preguntas?