

## Laboratorio QoS en SDN – V3

### Contenido

Laboratorio – SDN con QoS.....	2
Objetivo Práctica Laboratorio. ....	2
Requisitos Laboratorio .....	2
Actividades Laboratorio .....	2
Topología Proyecto. ....	3
Instalación Floodlight con QoS.....	4
Configuración Red SDN .....	5
Configuración QoS en Floodlight.....	5
Configuración Topología en GNS 3.....	6
Pruebas.....	7

## Laboratorio – SDN con QoS.

### Objetivo Práctica Laboratorio.

EXPERIMENTAR la operación de una red definida por software con políticas de Calidad de Servicio (QoS).

INTEGRAR una red definida por software con una red tradicional.

### Requisitos Laboratorio

Para el correcto desarrollo de este laboratorio es necesario contar con lo siguiente:

- Máquina virtual del Laboratorio - Preparación Ambiente.
- Mininet.
- Acceso a Internet.
- Máquina virtual controlador.
- Usuario y Servidor para pruebas.
- GNS 3.

### Actividades Laboratorio

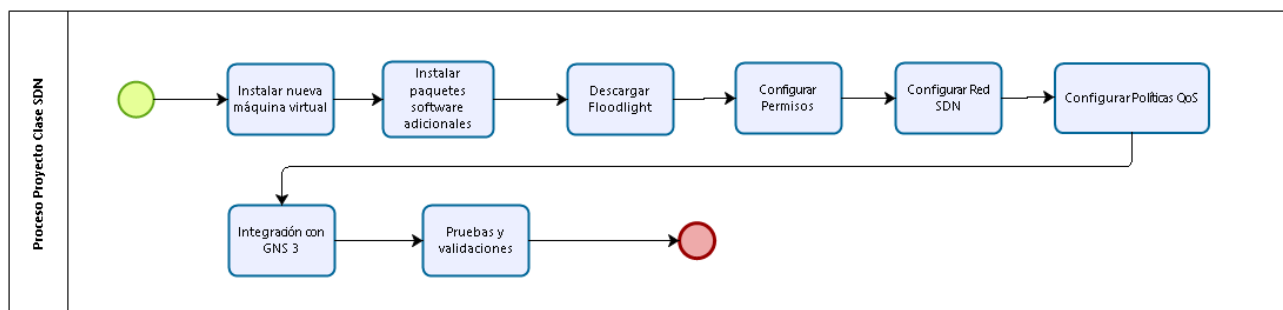


Diagrama 1. Proceso Proyecto

## Topología Proyecto.

Para el desarrollo del presente proyecto se propone utilizar como base la siguiente topología de red:

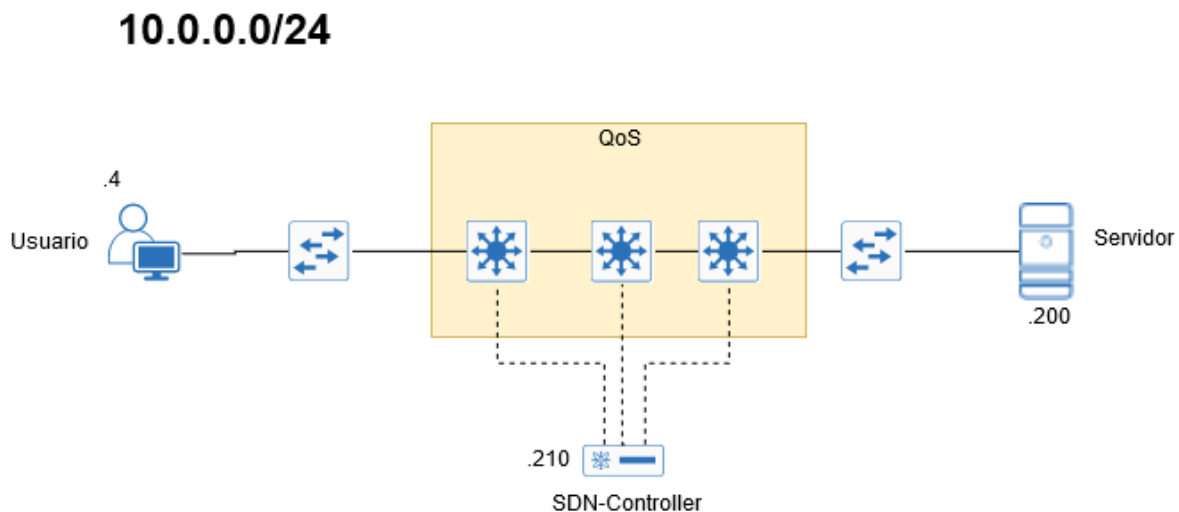


Figura 1. Topología Red Proyecto

Esta topología será simulada con el programa de emulación GNS 3, y así, poder experimentar un ambiente híbrido entre una arquitectura tradicional y una red definida por software. A continuación, se describe cada componente de la topología:

- **Red SDN:** Lo que se encuentra demarcado por el cuadro amarillo, corresponde a la red SDN con calidad de servicios. Se utilizan Open vSwitches gestionados por un controlador floodlight. Esta red SDN será simulada con la ayuda del programa Mininet.
- **Usuario:** El Usuario será una máquina virtual de Ubuntu 16.04 Desktop, quien se encargará de solicitar servicios al servidor.
- **Servidor:** El servidor será quien ofrezca diferentes servicios como FTP, HTTP o video streaming.
- **Switches Nivel 2:** Los switches nivel 2 serán switches tradicionales ofrecidos dentro de las herramientas del programa GNS 3.

Las políticas de Calidad de Servicio se basarán en la modificación de los flujos de tráfico utilizando el protocolo Open Flow.

## Instalación Floodlight con QoS.

Para el desarrollo de este laboratorio, nos basaremos en la siguiente guía del controlador floodlight: <https://floodlight.atlassian.net/wiki/spaces/floodlightcontroller/pages/1343504/How+to+implement+Quality+Of+Service+using+Floodlight>

Es importante tener en cuenta, que esté controlador debe instalarse en una máquina virtual diferente, considerando los siguientes recursos:

- **CPU:** 1 Núcleo.
- **Memoria RAM:** 2 GB.
- **Almacenamiento:** 20 GB.
- **Red:** Modo NAT/Conexión Puente.

Para evitar problemas a la hora de ejecutar este controlador SDN, deberemos detener el servicio del ovs-testcontroller instalado en prácticas anteriores:

```
sudo systemctl stop openvswitch-testcontroller
```

Antes de iniciar el proceso de instalación se deben instalar los siguientes componentes:

- Java JDK y JRE versión 1.7 (<https://do.co/2zdMOzb>)
- Curl

Iniciamos el proceso, descargando una copia del controlador Floodlight con el módulo QoS utilizando el siguiente comando:

```
$git clone https://github.com/wallnerrryan/floodlight-qos-beta.git
```

Para ejecutar el controlador floodlight se ejecutan los siguientes comandos:

```
$cd floodlight-qos-beta
```

```
$ant; ./floodlight.sh
```

Para acceder vía web al controlador Floodlight utilizamos la siguiente dirección:

```
http://Dir_IP_Floodlight:8080/ui/index.html
```

Para garantizar la correcta operación del modulo de Calidad de Servicio, se deben modificar los siguientes archivos:

### Modificar qospath2.py:

```
import sys
import os
```

```
import re
import time
#import simplejson #used to process policies and
encode/decode requests
import json as simplejson
import subprocess #spawning subprocesses
import argparse
```

### Modificar qosmanager.py

```
import sys
import os # for file handling
import httplib # basic HTTP library for HTTPS connections
import urllib # used for url-encoding during login request
#import simplejson # converts between JSON and python objects
import json as simplejson
import time # for dates in json
import argparse # more flexible argument parser for v2
```

### Configuración Red SDN

Para configurar la red SDN del proyecto, deberá utilizar la máquina virtual que se ha venido trabajando durante el semestre.

Ejecutar el siguiente comando para crear la red SDN:

```
sudo mn --topo linear 3 --switch ovsk -controller
remote,10.0.0.210:6633
```

Validar, que la red SDN se encuentre conectada con el controlador.

### Configuración QoS en Floodlight.

Ahora bien, ya tenemos nuestra red SDN controlada a través de Floodlight, podremos entonces realizar las configuraciones de calidad de servicio:

Primero debemos habilitar el módulo de QoS utilizando el siguiente comando:

```
sudo ./qosmanager2.py -e
```

Ahora, se deben agregar las colas a

```
sudo ./mininet-add-queues.py
```

Una vez creadas las colas, procedemos con la creación del flujo de tráfico que utilizará una política de calidad de servicio. Esta política utiliza 3 colas diferentes, para configurar el flujo y las colas utilizamos el siguiente comando:

```
./qospath2.py -a -S 10.0.0.5 -D 10.0.0.5 -N 2Mbit_1-5 -J '{"eth-type":"0x0800","queue":"2"}'
```

- ❓ Realice la configuración de diferentes colas modificando el parámetro “queue” entre los valores 1,2,3. ¿Qué efecto puede apreciar en la red SDN al aplicar la configuración de cada una de las colas?

Puede validar los servicios que están operando en el controlador con el siguiente comando:

```
sudo ./qosmanager2.py --list -t services
```

Para validar la lista de políticas que están operando en el controlador, utilizar el siguiente comando:

```
sudo ./qosmanager2.py --list -t policies
```

Puede remover una política con el siguiente comando:

```
./qospath2.py -d -N 2Mbit_1-5
```

### Configuración Topología en GNS 3.

Para poder desarrollar este proyecto, se debe instalar la herramienta GNS 3. Considere las siguientes guías:

- Instalación GNS 3 en Windows: <https://youtu.be/2al6ipuJPLo>
- Instalación GNS 3 en Ubuntu: <https://youtu.be/dwCCDpZ93eQ>

Para Emular equipos Cisco en GNS 3 pueden utilizar los siguientes enlaces:

- Importar Máquinas Virtuales a GNS 3: <https://youtu.be/b-BPLD6FRHg>

Una vez tenga importadas las máquinas virtuales a la topología de GNS 3, considere la siguiente guía para conectar su topología de red SDN con el resto de la red:

<https://www.gns3.com/news/article/sdn-101-mininet-openflow-and-gns>

Al finalizar deberá contar con una conexión similar a la presentada en la siguiente captura:

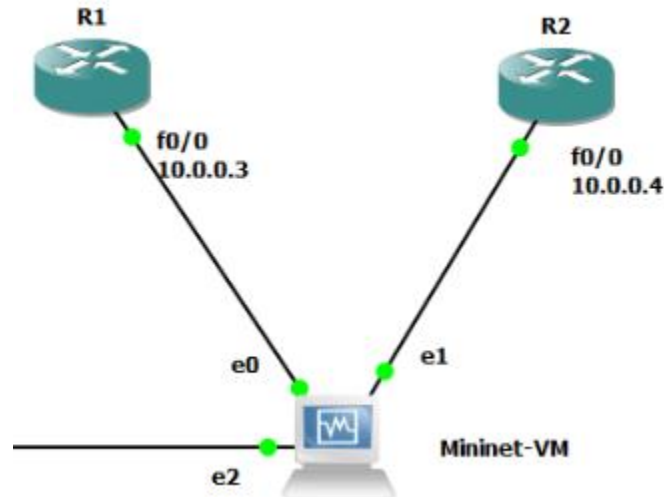


Figura 2. Imagen referencia conectividad Mininet.

## Pruebas

Una vez haya terminado de construir toda la topología híbrida, incluyendo todos los elementos, procedemos a realizar las diferentes validaciones de la red.

Para poder tener una simulación más acercada a la realidad y apreciar el funcionamiento de las políticas QoS, es importante congestionar la red. Para esta labor, se utiliza el programa iperf. Instale iperf en cada uno de los extremos de la comunicación (Usuario y Servidor) y ejecute los siguientes comandos:

### Habilitar Servidor iPerf

```
iperf -s -P 0 -i 1 -p 5001 -f m
```

### Habilitar Cliente iPerf

```
iperf -c 10.0.0.200 -P 1 -i 1 -p 5001 -f m -t 900 -T 1
```

Apóyese en los datos obtenidos con la herramienta iperf y en la transmisión del archivo por FTP y analice los siguientes datos:

- ❓ Ancho de banda y Retardo de la red híbrida **sin** aplicar políticas de calidad de servicio. Deberán presentar una gráfica que muestre el comportamiento de cada métrica (ancho de banda y retardo) al menos en 15 minutos de simulación. Deberán interpretar cada una de las gráficas.
- ❓ Ancho de banda y Retardo de la red híbrida **con** aplicar políticas de calidad de servicio. Deberán presentar una gráfica que muestre el comportamiento de cada métrica (ancho de

banda y retardo) al menos en 10 minutos de simulación. Deberán realizar la simulación por cada una de las colas configuradas (1,2,3) e interpretar cada una de las gráficas.

- ❓ Contraste ambos comportamientos y escriba sus conclusiones respecto al impacto de implementar políticas de Calidad de Servicio.