

# Laboratorio 3

## Redes de Computadores

Profesor: Erika Rosas Olivos  
Ayudantes: Ignacio Cofré  
Martín Crisóstomo  
Felipe Montero  
Benjamín Riquelme

Julio 2020

### 1. Objetivos

- Familiarizarse con la herramienta emuladora de redes *Mininet*
- Experimentar con SDN (redes definidas por software) y OpenFlow

### 2. Introducción

Las redes definidas por software se caracterizan por ser administradas de una manera eficiente, dejando toda la carga de armado de tablas de ruta a un controlador capaz de monitorear los switches de dicha red, permitiendo una administración flexible y cuyos problemas sean fáciles de diagnosticar, centralizando el conocimiento de la red a dicho controlador. Además, dicha flexibilidad permite tener una mayor seguridad en la red, creando “firewalls” altamente personalizables al denegar ciertos tipos de “frames”. *OpenFlow* es un protocolo SDN el cual permite al controlador definir las tablas de rutas de cada switch en base a reglas aplicadas en las siguientes capas: *link*, *network* y *transporte*. Estas decisiones son distribuidas a cada uno de los switches de la red para que puedan mover los frames de la manera esperada por el administrador de la red.

### 3. Laboratorio

El laboratorio consiste en configurar 2 redes con topologías distintas, donde sus switches empleen OpenFlow para la distribución de *frames*. Para lograr este objetivo, se debe emular la red a través de la herramienta *Mininet*.

#### 3.1. Red 1: Anillo simple

Se debe crear una red compuesta de 6 hosts y 3 switches, de tal manera que cada switch este conectado directamente a 2 hosts y los otros 2 switches. Cada host debe estar conectado solamente a un switch. Se presenta la topología en la siguiente imagen.

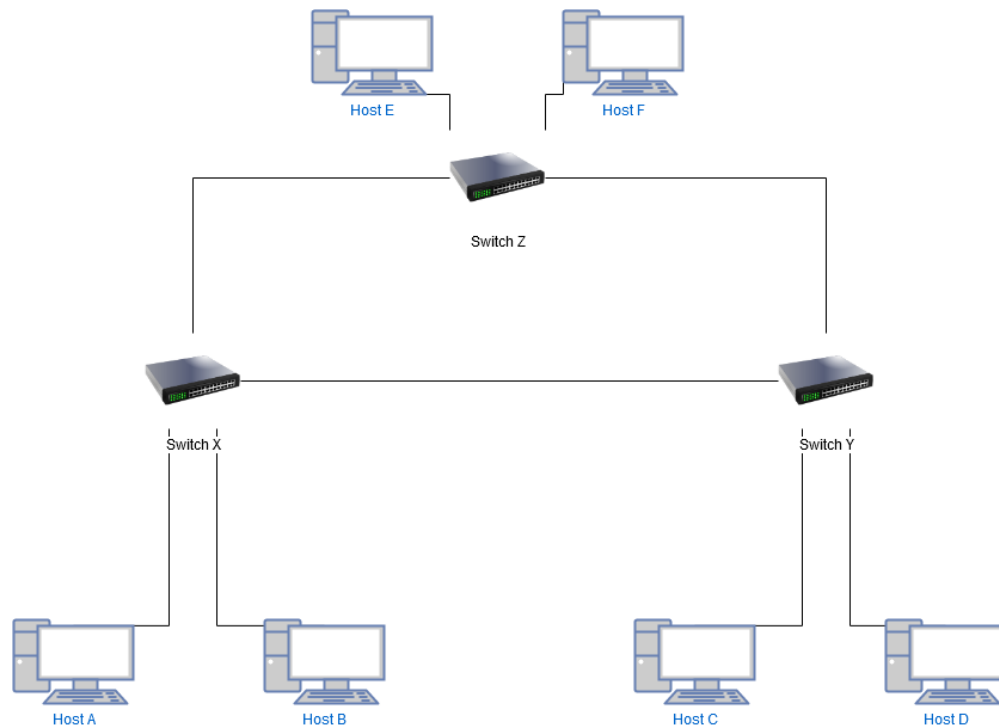


Figura 1: Topología de la red 1

Se debe configurar, considerando el protocolo OpenFlow, el controlador y las tablas de flujo de tal manera que todos los host puedan comunicarse entre sí, es decir, realizar exitosamente un **ping**.

Además, se debe experimentar que ocurre al eliminar uno de los enlaces entre los switches. Comente lo ocurrido en el README o un documento PDF si lo estima conveniente. *Hint: use Wireshark para saber que hace el controlador*

Posteriormente, modifique las configuraciones para que la red tenga el siguiente comportamiento:

- Los *frames* solo pueden circular entre los switches en un solo sentido, horario o anti-horario.
- Maximizar el uso del enlace entre los switches X e Y.

### 3.2. Red 2: Dos Caminos

Se debe crear una red compuesta por 7 host y 7 switches. Cada par de host debe estar conectado a un switch distinto, el cual a la vez se conectará a otro switch, el séptimo host, el cual actuará como servidor HTTP, estará conectado a un único switch, y todos los switch se conectarán de la siguiente manera:

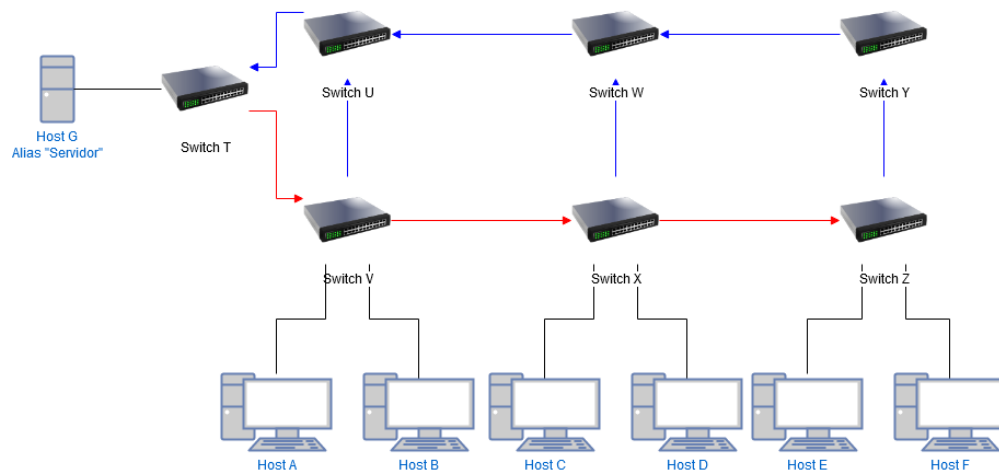


Figura 2: Topología de la red 2

El color y dirección de las flechas indican la dirección que deben tomar los frames: los cables negros son bidireccionales, mientras que los cables rojos y azules serán unidireccionales. Si el frame corresponde a una request al servidor HTTP, debe ser transmitido a través de los cables azules, mientras que las respuestas hechas por el servidor deben avanzar a través de los cables rojos.

Se debe configurar las tablas de flujo de tal manera que todos los host puedan comunicarse con el servidor HTTP, es decir, realizar exitosamente un `wget`, y siguiendo las reglas mencionadas en esta sección, además de bloquear todo intento de comunicación host-host y levantar un "firewal" en el servidor que bloquee todo frame que no sea dirigido al servicio HTTP.

## 4. Consideraciones

- Solo puede existir un controlador SDN en cada red.
- Se debe agregar reglas que eviten que hosts ajenos interactúen con la red. Entiéndase host ajeno a un host no considerado en la topología inicial
- Si el destino de un frame es un host ajeno, el switch debe botar dicho frame.
- Se da libertad de elección de la plataforma del controlador, mientras sea compatible con Mininet. Única excepción es *NOX*, ya que el proyecto se encuentra actualmente abandonado y no debe ser usado.
- Los labels de hardware presentados en las figuras son sugerencias. Sin embargo, debe informar en el README los labels usados para cada componente. Ejemplo: los host de la A a la F están rotulados del 1 al 6.
- Consultas sobre la tarea se deben realizar en aula o enviar un correo a los ayudantes:

[martin.crisostomo@sansano.usm.cl](mailto:martin.crisostomo@sansano.usm.cl)

[benjamin.riquelme@sansano.usm.cl](mailto:benjamin.riquelme@sansano.usm.cl)

## 5. Reglas de entrega

- La tarea se realiza en **grupos de 2 personas**.

- La fecha de entrega es el día **viernes 7 de agosto de 2020**.
- La tarea debe usar la versión 2.2.2 de Mininet, corriendo en Ubuntu LTS.
- La entrega debe realizarse a través de Aula, en un archivo comprimido **ZIP** o **TAR GZ**, cuyo nombre debe indicar el rol sin dígito verificador de los integrantes. Ejemplo: **T3-Rol1-Rol2.zip**. El fichero debe contener los archivos necesarios para reconstruir las redes creadas:
  - Topología: Script de Python o TXT con comandos de terminal/Mininet CLI
  - Controlador: Código o instrucciones para configurar correctamente el controlador. Para la red 1 se consideran las 3 instancias de controlador, mientras que para la red 2 se considera solo una.
- Debe entregar un **README** con nombre y rol de cada integrante, el cómo usar los archivos entregados para levantar la red y consideraciones especiales.
- Cada día de atraso se penalizará con un descuento de **10 puntos** hasta los 3 días, Posterior a dicho tiempo se evaluará con **nota 0**.
- Copias serán evaluadas con **nota 0**