



75.43

Introducción a Sistemas Distribuidos

Trabajo Práctico 3

Software Defined Networks

1. Objetivo

Este trabajo práctico se plantea como objetivo el estudio, la comprensión y la puesta en práctica de los conceptos y herramientas necesarias para la implementación de redes definidas por software (SDN).

Para lograr esto, el alumno deberá ser capaz de construir y simular un pequeño datacenter bajo la topología Fat-Tree, donde se pedirá utilizar OpenFlow para poder hacer uso de múltiples enlaces en simultáneo.

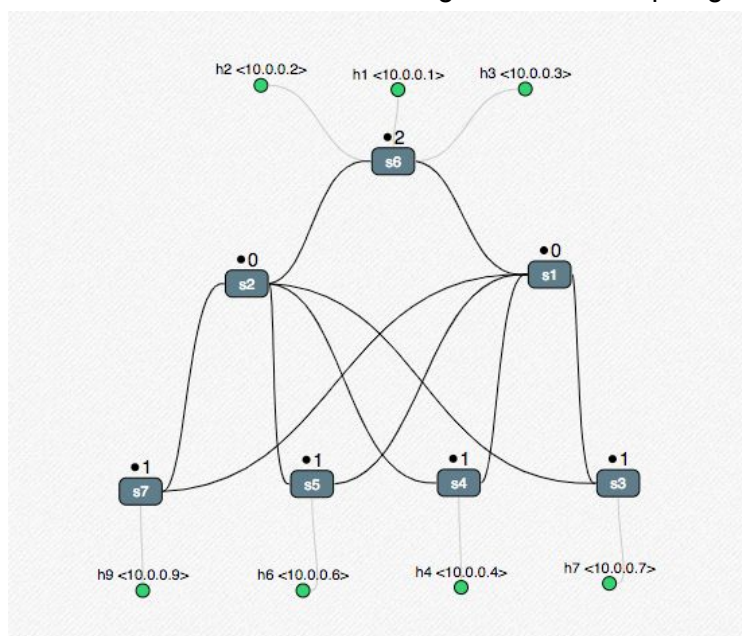
2. Requerimientos

Para poder realizar la emulación del datacenter, el grupo deberá describir la topología solicitada utilizando Mininet.

La topología a simular es conocida como Fat-Tree topology y consiste en armar un grafo de routers, partiendo de una sola raíz. Esta topología tiene por característica que para un nivel n , consta de 2^n routers, y además cada router en el nivel n tiene un enlace con todos los routers del nivel $n + 1$.

Para realizar las distintas simulaciones que detallaremos más adelante, el router en la raíz deberá estar conectado con 3 hosts y cada router hoja (perteneciente al último nivel) deberá estar conectado a 1 host.

A continuación se muestra un diagrama de una topología Fat-Tree de 3 niveles.



Para la simulación, la generación de la topología deberá ser parametrizable en la cantidad de niveles a generar.

ECMP

Como se menciona en la RFC2991, ante la posibilidad de que un router tenga más de un camino con el mismo costo para alcanzar un destino, cada **flujo** debe ser siempre enviado por el mismo camino.

Esta técnica se llama Equal Cost Multiple Path (**ECMP**), y también es sumamente popular a la hora del balance de carga a nivel de capa de enlace dentro de los datacenters.

Nuestro objetivo aquí será poder llevar a cabo esta técnica de manera tal de hacer uso de todos los enlaces de manera equitativa.

La figura anterior muestra cómo entre h1 y h5 existen dos caminos con exactamente el mismo costo, por lo cual, dado un flujo, el camino debe ser el mismo.

Sin embargo, ante flujos distintos entre h1 y h5 se puede hacer uso del camino restante.

Nota de implementación

Cuando configuramos un router o switch openflow mediante pox, cuando éste recibe un paquete, hay dos casos posibles:

- Tiene una regla en su tabla que matchea el paquete, con lo que lo dirige a la interfaz que corresponda (operación muy rápida).
- No coincide ninguna regla. En este caso, se le pregunta al controlador cuál interfaz corresponde (operación más lenta).

De esto surge una implicación de performance: si configuramos nuestra implementación para que siempre vaya al controlador, será muy lenta. Por lo tanto, debe escribirse la tabla de los switches/routers periódicamente, para que la próxima vez que un paquete similar pase por este, pueda obviarse el llamado al controlador.

3. Herramientas

El framework para poder ejecutar mininet y OpenFlow será provisto por la cátedra, a través del siguiente repositorio: <https://github.com/7543distrofiuba/tp3-framework>

Este framework utiliza docker para aislar la instalación de mininet del resto del entorno del usuario. Para configurarlo, así como poder correr la topología de ejemplo (que usa STP para controlar los switches) de debe hacer lo indicado en el README. Allí además tendremos indicaciones de cómo utilizar las herramientas y correr los programas de prueba.

4. Condiciones de aprobación

Los grupos, de **máximo 4 personas, mínimo 3**, deberán entregar el código completo del trabajo práctico, en **formato ZIP**, junto con un informe de **máximo 5 páginas**, en **formato PDF**, donde deberán incluir las siguientes secciones.

Implementación

La topología a desarrollar deberá ser configurable, siendo el parámetro, la cantidad de niveles del árbol. La raíz del mismo es el nivel 0, y las hojas el nivel $2^{(altura\ del\ arbol - 1)}$.

La raíz debe tener conectada 3 hosts que funcionarán como clientes de nuestros datacenter, cada una de las hojas a su vez, tendrán conectadas un host cada una, que funcionará como proveedor de contenido.

El controlador debe ser capaz de distribuir el tráfico uniformemente considerando todos los conceptos aprendidos hasta el momento. Para ello debe indicar al dispositivo de que forma llenar su CAM table, haciendo uso de la técnica ECMP y teniendo en cuenta el concepto de flujo antes visto.

Informe

- Introducción.
- Comparación de SDNs y Openflow contra protocolos de ruteo clásicos
- Suposiciones y/o asunciones tomadas para realizar el trabajo.
- Dificultades encontradas.
- Conclusión

Aclaración: Las páginas excedentes no serán consideradas.

5. Entrega

La entrega se hará a través del campus de la misma forma que los ejercicios de la semana. La fecha máxima para subir el código y el informe será el **domingo 19/07/2019 a las 19.00hs**. Cualquier entrega fuera de término no será considerada como tal.