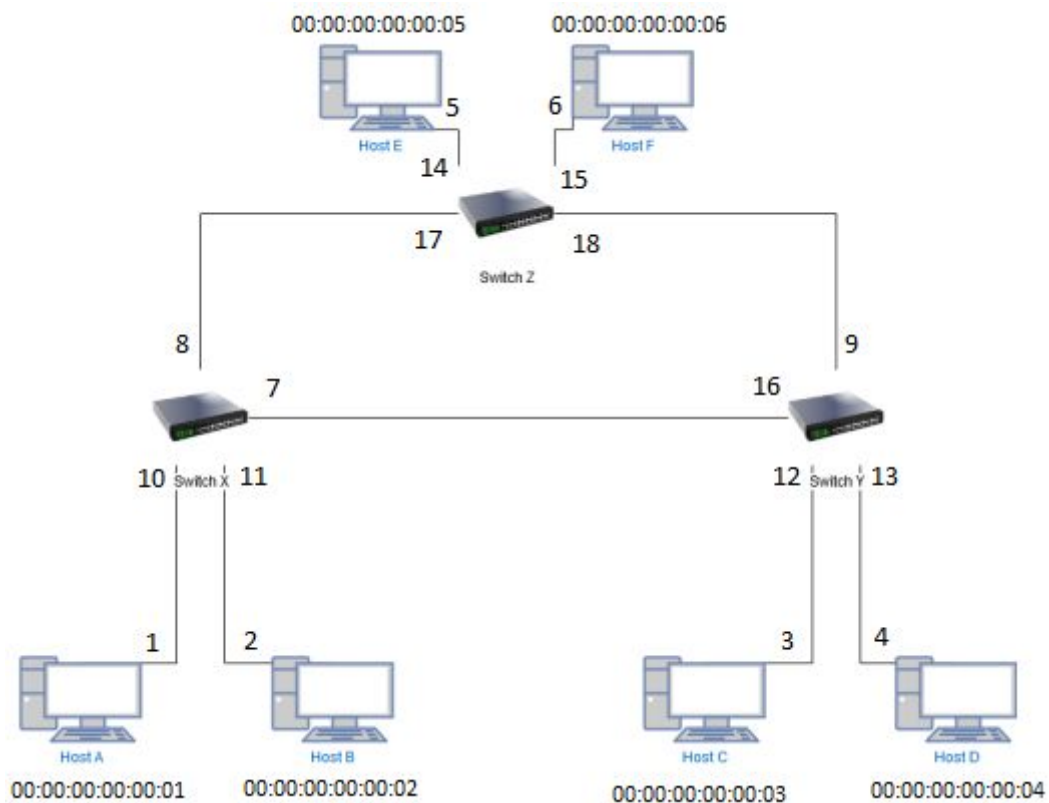


Laboratorio 3

Integrantes: Ignacio Aedo, ROL 201773556-2

Ethiel Carmona, ROL 201773533-3

RED 1: Anillo simple



La topología se realizó en el archivo *ethielcita.py*.

En el presente informe, se consideraron los flujos de los archivos *l2_learning.py*, *aedita.py* y *aeditamax.py* con la configuración de OpenFlow.

Topología con flujo brindado por *l2_learning.py*

Las instrucciones para correr nuestro trabajo son (teniendo instalado mininet y pox):

- Donde se encuentre el archivo *ethielcita.py* se debe ejecutar la siguiente línea:
`sudo mn --custom ethielcita.py --topo MyTopo --controller remote --switch ovsk --mac`

- Las líneas de comando se deben ejecutar en la carpeta de afuera donde se encuentre instalado pox. El archivo *l2_learning.py* debe estar guardado en la carpeta forwarding de pox (pox->pox->forwarding), ejecutar la línea de comando: `python2.7 pox/pox.py --verbose openflow.spanning_tree --no-flood --hold-down openflow.discovery forwarding.l2_learning`

Para probar que la topología funciona, ejecutamos el comando *pingall*:

```
mininet> pingall
*** Ping: testing ping reachability
hA -> hB hC hD hE hF
hB -> hA hC hD hE hF
hC -> hA hB hD hE hF
hD -> hA hB hC hE hF
hE -> hA hB hC hD hF
hF -> hA hB hC hD hE
*** Results: 0% dropped (30/30 received)
```

Eliminación enlace entre Switch X y Switch Z con *l2_learning.py*

Se ejecuta el comando mostrado en la imagen:

```
mininet> link s1 s3 down
mininet>
```

Probamos haciendo un *pingall* entre el host A y el host E y notamos que se pierden paquetes:

```
mininet> pingall
*** Ping: testing ping reachability
hA -> hB hC hD X X
hB -> hA hC hD X X
hC -> hA hB hD hE hF
hD -> hA hB hC hE hF
hE -> hA hB hC hD hF
hF -> hA hB hC hD hE
*** Results: 13% dropped (26/30 received)
mininet>
```

Realizamos *pingall* nuevamente:

```
mininet> pingall
*** Ping: testing ping reachability
hA -> hB hC hD hE hF
hB -> hA hC hD hE hF
hC -> hA hB hD hE hF
hD -> hA hB hC hE hF
hE -> hA hB hC hD hF
hF -> hA hB hC hD hE
*** Results: 0% dropped (30/30 received)
mininet>
```

Se puede observar que esta segunda vez no hay pérdida de mensajes. Esto se debe al aprendizaje de nuevas rutas que se implementó en `I2_learning`.

Topología con flujo brindado por `aedita.py`

Se consideró un flujo antihorario. Además, los comandos a utilizar (***directorios indicados en `I2_learning.py`***) son los siguientes:

- `sudo mn --custom ethielcita.py --topo MyTopo --controller remote --switch ovsk --mac`
- `python2.7 pox/pox.py --verbose openflow.spanning_tree --no-flood --hold-down openflow.discovery forwarding.aedita`

Realizamos un *pingall* para verificar que nuestra topología funciona:

```

*** Adding hosts:
hA hB hC hD hE hF
*** Adding switches:
s1 s2 s3
*** Adding links:
(hA, s1) (hB, s1) (hC, s2) (hD, s2) (hE, s3) (hF, s3) (s1, s2) (s1, s3) (s2, s3)

*** Configuring hosts
hA hB hC hD hE hF
*** Starting controller
c0
*** Starting 3 switches
s1 s2 s3 ...
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
hA -> hB hC hD hE hF
hB -> hA hC hD hE hF
hC -> hA hB hD hE hF
hD -> hA hB hC hE hF
hE -> hA hB hC hD hF
hF -> hA hB hC hD hE
*** Results: 0% dropped (30/30 received)
mininet>

```

```

nachito@nachito-VirtualBox: ~/Escritorio/mininet
nachito@nachito-VirtualBox:~/Escritorio/mininet$ python2.7 pox/pox.py --verbose
openflow.spanning_tree --no-flood --hold-down openflow.discovery forwarding.aedita
POX 0.3.0 (dart) / Copyright 2011-2014 James McCauley, et al.
DEBUG:openflow.spanning_tree:Spanning tree component ready
DEBUG:core:POX 0.3.0 (dart) going up...
DEBUG:core:Running on CPython (2.7.18rc1/Apr 7 2020 12:05:55)
DEBUG:core:Platform is Linux-5.4.0-42-generic-x86_64-with-Ubuntu-20.04-focal
INFO:core:POX 0.3.0 (dart) is up.
DEBUG:openflow.of_01:Listening on 0.0.0.0:6633
INFO:openflow.of_01:[00-00-00-00-00-01 2] connected
DEBUG:openflow.discovery:Installing flow for 00-00-00-00-00-01
DEBUG:openflow.spanning_tree:Disabling flooding for 5 ports
DEBUG:forwarding.aedita:Connection [00-00-00-00-00-01 2]
INFO:openflow.of_01:[00-00-00-00-00-02 4] connected
DEBUG:openflow.discovery:Installing flow for 00-00-00-00-00-02
DEBUG:openflow.spanning_tree:Disabling flooding for 5 ports
DEBUG:forwarding.aedita:Connection [00-00-00-00-00-02 4]
INFO:openflow.of_01:[00-00-00-00-00-03 3] connected
DEBUG:openflow.discovery:Installing flow for 00-00-00-00-00-03
DEBUG:openflow.spanning_tree:Disabling flooding for 5 ports
DEBUG:forwarding.aedita:Connection [00-00-00-00-00-03 3]
INFO:openflow.discovery:link detected: 00-00-00-00-00-01.8 -> 00-00-00-00-00-03.
17

```

Podemos observar con Openflow corriendo que todos los pings posibles funcionan.

Topología con flujo brindado por aeditamax.py

Los comandos a utilizar (*directorios indicados en l2_learning.py*) son los siguientes:

- `sudo mn --custom ethielcita.py --topo MyTopo --controller remote --switch ovsk --mac`
- `python2.7 pox/pox.py --verbose openflow.spanning_tree --no-flood --hold-down openflow.discovery forwarding.aeditamax`

Al hacer un pingall se puede observar que no hay pérdida de paquetes.

```
mininet> pingall
*** Ping: testing ping reachability
hA -> hB hC hD hE hF
hB -> hA hC hD hE hF
hC -> hA hB hD hE hF
hD -> hA hB hC hE hF
hE -> hA hB hC hD hF
hF -> hA hB hC hD hE
*** Results: 0% dropped (30/30 received)
mininet>
```