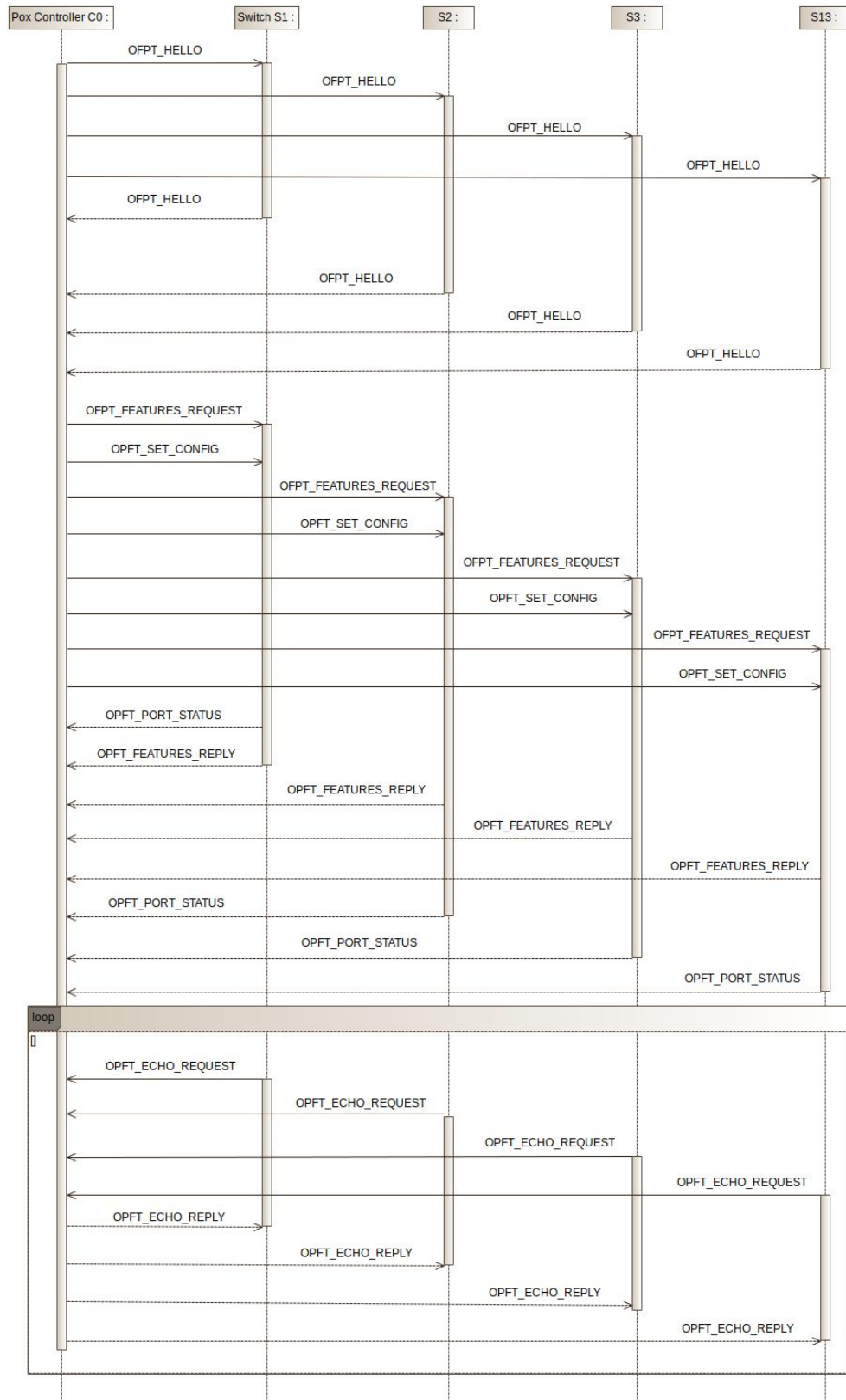


# TALLER DE OPENFLOW

Presentado por: Jonathan Aldana C.

1. Hacer el diagrama de secuencia cuando se crea la red en Mininet.  
R/

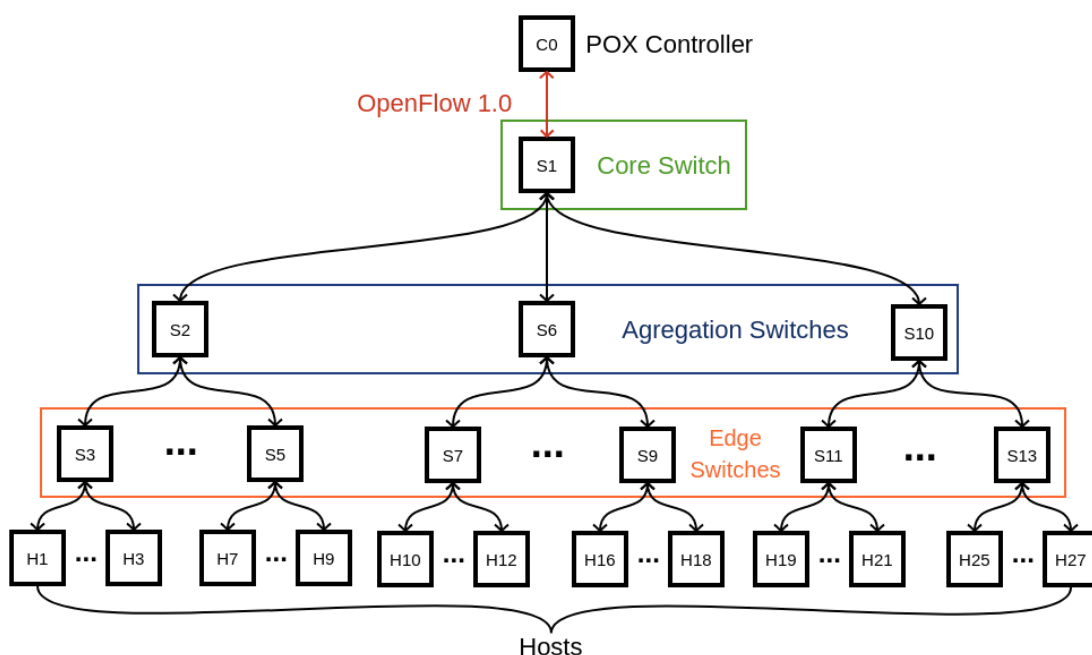


Cuando se crea la red y los enlaces ya están establecidos entre cada uno de los switches y el controlador, dicho controlador (**C0**) comienza a enviar un **OPFT\_HELLO** a cada uno de los switches de la red creada con el fin de intercambiar la versión de OpenFlow más alta que puede soportar para asegurarse de que la negociación entre el controlador y los Switches es posible, en este caso el **C0** envió 13 mensajes **OPFT\_HELLO** y cada uno de los Switch le respondió con un **OPFT\_HELLO** a **C0**.

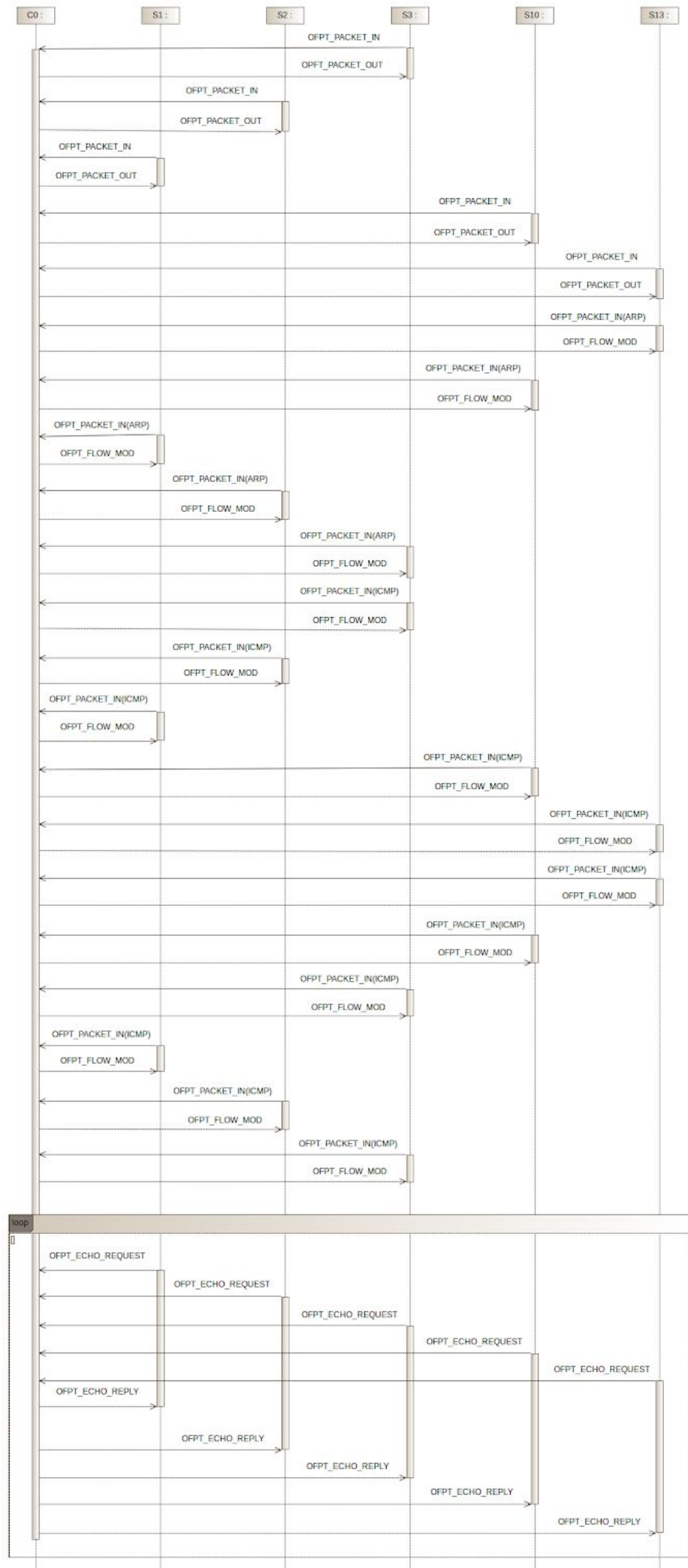
Cuando **C0** ha recibido la respuesta del **OPFT\_HELLO** ya procede a enviar a cada uno de los switch los mensajes **OPFT\_FEATURES\_REQUEST** (para solicitarle a cada uno de los switch las capacidades que tiene) y **OPFT\_SET\_CONFIG** (para consultar y establecer las propiedades para el manejo del procesamiento de paquetes en cada uno de los switch), luego cuando cada switch recibe dichos mensajes del controlador, para el caso del **OPFT\_FEATURES\_REQUEST** el switch le responde con un **OPFT\_FEATURES\_REPLY** con todas las capacidades con las que cuenta como capacidades, acciones, puertos, etc; luego de esto el switch envía un **OPFT\_PORTSTATUS** para avisarle a **C0** que la configuración y los cambios en el puerto fueron realizados correctamente.

Por último cuando todo ya esta configurado correctamente cada uno de los switches le envía a **C0** el mensaje **OPFT\_ECHO\_REQUEST** para intercambiar información sobre la latencia, ancho de banda y la vitalidad del enlace y así cuando **C0** responde con un **OPFT\_ECHO\_REPLY** a cada switch, se asegura que la red funciona correctamente. Este ciclo es infinito hasta que se uno de los Host solicita algo a la red.

2. Hacer el diagrama de secuencia cuando se hace ping entre el Host 1 y Host 27.  
R/



Teniendo en cuenta la topología de la red creada se procede a hacer ping entre H1 y H27, el diagrama de secuencia de paquetes OpenFlow obtenido es el siguiente:



Cuando se ejecuta el comando **h1 ping h27** en mininet, el primer paquete OpenFlow que se genera ocurre inmediatamente en el switch 3 (**S3**), donde está asociado **H1**, con el mensaje **OPFT\_PACKET\_IN** en donde le pregunta al controlador **C0** qué hacer con el paquete que recibió, este le responde con un **OPFT\_PACKET\_OUT** diciendole que lo replique a los otros switch para encontrar **H27**, por lo que desde aquí comienza una cadena de mensajes **OPFT\_PACKET\_IN** y **OPFT\_PACKET\_OUT** entre todos los switches y el controlador para distribuir las direcciones Mac de la red, hasta que dicho intercambio de mensajes llega al Switch 13 (**S13**), donde casi inmediatamente luego de recibir el **OPFT\_PACKET\_OUT** del controlador, **S13** envía un **OPFT\_PACKET\_IN(ARP)** preguntándole a **C0** qué hacer para ejecutar el protocolo **ARP** ya que el tiene asociado a **H27** y ahora necesita encontrar a **H1**, **C0** le responde a **S13** con **OPFT\_FLOW\_MOD** indicándole por qué puerto debe mandar dicho paquete, esto se repite con **S10**, **S1**, **S2** y **S3**, cuando llega a **S3** este ya tiene la información por donde podría enviar el paquete **ICMP** (En los 5 switches ya se tiene una tabla de flujo) por ende inicializa un mensaje para **C0** con **OPFT\_PACKET\_IN(ICMP)** para preguntarle por qué puerto debe enviar dicho paquete y **C0** le responde con un **OPFT\_FLOW\_MOD** para modificar la tabla de flujos y decirle por qué puerto debe enviar dicho paquete, esto se repite hasta que llega a **S13** y se repite con el mensaje de respuesta por parte de **H27**, de esta manera se construye el camino del ping y luego todos los Switches y el controlador entran en el ciclo de **OPFT\_ECHO\_REQUEST** y **OPFT\_ECHO\_REPLY**.

- A partir del punto 2 revisar los switches donde se crearon flujos y explicar.  
R/

Teniendo en cuenta el punto anterior, se muestra a continuación los 5 switches implicados en el mantenimiento del camino generado por el ping entre **H1** y **H27**:

```

"Node: s2" (root)
root@mininet-vw:~# dctl dump-flows top:127.0.0.1:6655
stats_reply (xid=0x34aebcd1): flags:none type=1(flow)
cookie=0, duration_sec=16s, duration_nsec=495000000, table_id=0, priority=655
35, n_packets=1, n_bytes=42, idle_timeout=60, hard_timeout=0, arp, in_port=4, dl_vlan=0xffff, dl_src=00:00:00:00:00:01, dl_dst=00:00:00:00:00:01, nw_src=10.0.0.27, nw_dst=10.0.0.1, nw_proto=1, actions=output:1
cookie=0, duration_sec=16s, duration_nsec=450000000, table_id=0, priority=655
35, n_packets=1, n_bytes=42, idle_timeout=60, hard_timeout=0, arp, in_port=1, dl_vlan=0xffff, dl_src=00:00:00:00:00:01, dl_dst=00:00:00:00:00:01, nw_src=10.0.0.1, nw_dst=10.0.0.27, nw_proto=2, actions=output:4
cookie=0, duration_sec=21s, duration_nsec=633000000, table_id=0, priority=655
35, n_packets=1, n_bytes=42, idle_timeout=60, hard_timeout=0, arp, in_port=4, dl_vlan=0xffff, dl_src=00:00:00:00:00:01, dl_dst=00:00:00:00:00:01, nw_src=10.0.0.27, nw_dst=10.0.0.1, nw_proto=2, actions=output:1
cookie=0, duration_sec=21s, duration_nsec=498000000, table_id=0, priority=655
35, n_packets=15, n_bytes=1470, idle_timeout=60, hard_timeout=0, icmp, in_port=4, dl_vlan=0xffff, dl_src=00:00:00:00:00:01, dl_dst=00:00:00:00:00:01, nw_src=10.0.0.27, nw_dst=10.0.0.1, nw_tos=0x00, icmp_type=8, icmp_code=0, actions=output:1
cookie=0, duration_sec=21s, duration_nsec=588000000, table_id=0, priority=655
35, n_packets=15, n_bytes=1470, idle_timeout=60, hard_timeout=0, icmp, in_port=1, dl_vlan=0xffff, dl_src=00:00:00:00:00:01, dl_dst=00:00:00:00:00:01, nw_src=10.0.0.1, nw_dst=10.0.0.27, nw_tos=0x00, icmp_type=8, icmp_code=0, actions=output:4
root@mininet-vw:~# dctl dump-flows top:127.0.0.1:6655
stats_reply (xid=0xc2c27bcd): flags:none type=1(flow)

"Node: s3" (root)
root@mininet-vw:~# dctl dump-flows top:127.0.0.1:6656
stats_reply (xid=0xdcb4cd1): flags:none type=1(flow)
cookie=0, duration_sec=36s, duration_nsec=513000000, table_id=0, priority=655
35, n_packets=1, n_bytes=42, idle_timeout=60, hard_timeout=0, arp, in_port=4, dl_vlan=0xffff, dl_src=00:00:00:00:00:01, dl_dst=00:00:00:00:00:01, nw_src=10.0.0.27, nw_dst=10.0.0.1, nw_proto=1, actions=output:1
cookie=0, duration_sec=36s, duration_nsec=491000000, table_id=0, priority=655
35, n_packets=1, n_bytes=42, idle_timeout=60, hard_timeout=0, arp, in_port=1, dl_vlan=0xffff, dl_src=00:00:00:00:00:01, dl_dst=00:00:00:00:00:01, nw_src=10.0.0.1, nw_dst=10.0.0.27, nw_proto=2, actions=output:4
cookie=0, duration_sec=41s, duration_nsec=651000000, table_id=0, priority=655
35, n_packets=1, n_bytes=42, idle_timeout=60, hard_timeout=0, arp, in_port=4, dl_vlan=0xffff, dl_src=00:00:00:00:00:01, dl_dst=00:00:00:00:00:01, nw_src=10.0.0.27, nw_dst=10.0.0.1, nw_proto=2, actions=output:1
cookie=0, duration_sec=41s, duration_nsec=515000000, table_id=0, priority=655
35, n_packets=15, n_bytes=1470, idle_timeout=60, hard_timeout=0, icmp, in_port=4, dl_vlan=0xffff, dl_src=00:00:00:00:00:01, dl_dst=00:00:00:00:00:01, nw_src=10.0.0.27, nw_dst=10.0.0.1, nw_tos=0x00, icmp_type=8, icmp_code=0, actions=output:1
cookie=0, duration_sec=41s, duration_nsec=520000000, table_id=0, priority=655
35, n_packets=15, n_bytes=1470, idle_timeout=60, hard_timeout=0, icmp, in_port=1, dl_vlan=0xffff, dl_src=00:00:00:00:00:01, dl_dst=00:00:00:00:00:01, nw_src=10.0.0.1, nw_dst=10.0.0.27, nw_tos=0x00, icmp_type=8, icmp_code=0, actions=output:4
root@mininet-vw:~# dctl dump-flows top:127.0.0.1:6656
stats_reply (xid=0x324d95c9): flags:none type=1(flow)

"Node: s10" (root)
root@mininet-vw:~# dctl dump-flows top:127.0.0.1:6663
stats_reply (xid=0xd38c3e44): flags:none type=1(flow)
root@mininet-vw:~# dctl dump-flows top:127.0.0.1:6663
stats_reply (xid=0x6724b433): flags:none type=1(flow)
cookie=0, duration_sec=1s, duration_nsec=372000000, table_id=0, priority=655
5, n_packets=1, n_bytes=42, idle_timeout=60, hard_timeout=0, arp, in_port=3, dl_vlan=0xffff, dl_src=00:00:00:00:00:01, dl_dst=00:00:00:00:00:01, nw_src=10.0.0.27, nw_dst=10.0.0.1, nw_proto=1, actions=output:1
cookie=0, duration_sec=1s, duration_nsec=588000000, table_id=0, priority=655
5, n_packets=1, n_bytes=42, idle_timeout=60, hard_timeout=0, arp, in_port=4, dl_vlan=0xffff, dl_src=00:00:00:00:00:01, dl_dst=00:00:00:00:00:01, nw_src=10.0.0.1, nw_dst=10.0.0.27, nw_proto=2, actions=output:1
cookie=0, duration_sec=5s, duration_nsec=378000000, table_id=0, priority=655
5, n_packets=7, n_bytes=686, idle_timeout=60, hard_timeout=0, icmp, in_port=3, dl_vlan=0xffff, dl_src=00:00:00:00:00:01, dl_dst=00:00:00:00:00:01, nw_src=10.0.0.27, nw_dst=10.0.0.1, nw_tos=0x00, icmp_type=8, icmp_code=0, actions=output:4
cookie=0, duration_sec=5s, duration_nsec=320000000, table_id=0, priority=655
5, n_packets=5, n_bytes=588, idle_timeout=60, hard_timeout=0, icmp, in_port=1, dl_vlan=0xffff, dl_src=00:00:00:00:00:01, dl_dst=00:00:00:00:00:01, nw_src=10.0.0.1, nw_dst=10.0.0.27, nw_tos=0x00, icmp_type=8, icmp_code=0, actions=output:3
root@mininet-vw:~# dctl dump-flows top:127.0.0.1:6663
stats_reply (xid=0x206271d): flags:none type=1(flow)
cookie=0, duration_sec=21s, duration_nsec=948000000, table_id=0, priority=655
35, n_packets=1, n_bytes=42, idle_timeout=60, hard_timeout=0, arp, in_port=4, dl_vlan=0xffff, dl_src=00:00:00:00:00:01, dl_dst=00:00:00:00:00:01, nw_src=10.0.0.27, nw_dst=10.0.0.1, nw_proto=2, actions=output:1

"Node: s1" (root)
root@mininet-vw:~# dctl dump-flows top:127.0.0.1:6654
stats_reply (xid=0x204d00a): flags:none type=1(flow)
cookie=0, duration_sec=25s, duration_nsec=122000000, table_id=0, priority=655
35, n_packets=1, n_bytes=42, idle_timeout=60, hard_timeout=0, arp, in_port=3, dl_vlan=0xffff, dl_src=00:00:00:00:00:01, dl_dst=00:00:00:00:00:01, nw_src=10.0.0.27, nw_dst=10.0.0.1, nw_proto=1, actions=output:1
cookie=0, duration_sec=30s, duration_nsec=261000000, table_id=0, priority=655
35, n_packets=1, n_bytes=42, idle_timeout=60, hard_timeout=0, arp, in_port=3, dl_vlan=0xffff, dl_src=00:00:00:00:00:01, dl_dst=00:00:00:00:00:01, nw_src=10.0.0.27, nw_dst=10.0.0.1, nw_proto=2, actions=output:1
cookie=0, duration_sec=25s, duration_nsec=590000000, table_id=0, priority=655
5, n_packets=1, n_bytes=42, idle_timeout=60, hard_timeout=0, arp, in_port=1, dl_vlan=0xffff, dl_src=00:00:00:00:00:01, dl_dst=00:00:00:00:00:01, nw_src=10.0.0.1, nw_dst=10.0.0.27, nw_proto=2, actions=output:3
cookie=0, duration_sec=30s, duration_nsec=126000000, table_id=0, priority=655
35, n_packets=15, n_bytes=1470, idle_timeout=60, hard_timeout=0, icmp, in_port=3, dl_vlan=0xffff, dl_src=00:00:00:00:00:01, dl_dst=00:00:00:00:00:01, nw_src=10.0.0.27, nw_dst=10.0.0.1, nw_tos=0x00, icmp_type=8, icmp_code=0, actions=output:1
cookie=0, duration_sec=30s, duration_nsec=194000000, table_id=0, priority=655
35, n_packets=15, n_bytes=1470, idle_timeout=60, hard_timeout=0, icmp, in_port=1, dl_vlan=0xffff, dl_src=00:00:00:00:00:01, dl_dst=00:00:00:00:00:01, nw_src=10.0.0.1, nw_dst=10.0.0.27, nw_tos=0x00, icmp_type=8, icmp_code=0, actions=output:3
root@mininet-vw:~# dctl dump-flows top:127.0.0.1:6654
stats_reply (xid=0x53145d): flags:none type=1(flow)

"Node: s13" (root)
root@mininet-vw:~# dctl dump-flows top:127.0.0.1:6656
stats_reply (xid=0xb55a0a5): flags:none type=1(flow)
cookie=0, duration_sec=24s, duration_nsec=388000000, table_id=0, priority=655
35, n_packets=1, n_bytes=42, idle_timeout=60, hard_timeout=0, arp, in_port=3, dl_vlan=0xffff, dl_src=00:00:00:00:00:01, dl_dst=00:00:00:00:00:01, nw_src=10.0.0.27, nw_dst=10.0.0.1, nw_proto=2, actions=output:1

```

Para este caso la tabla de flood de **S3** nos indica que para reenviar los paquetes **ICMP** que recibe de **H1** por el puerto 1, debe reenviarlo por el puerto 4, ya que en ese puerto de **S3** está la interfaz que conecta a **S2**, esto mismo ocurre en **S2** que recibe los datos de **S3** por el

puerto 1 y los reenvía a **S1** por el puerto 4, luego **S1** recibe los paquetes **S2** por el puerto 1 y los reenvía por el puerto 3 a **S10**, después **S10** tiene en su tabla de flujo que por el puerto 4 recibe los paquetes de **S1** y los reenvía a **S13** por el puerto 3 y por último **S13** recibe los paquetes de **S10** por el puerto 4 y los reenvía a **H27** por el puerto 3 para que **H27** los reciba por el puerto 0, así mismo cuando **H27** le responde el mensaje **ICMP** a **H1**. Para esta parte y se entendiera más fácil se asume que los puertos son equivalentes a las interfaces Ethernet de los Host y los Switches implicados en este intercambio.

4. Generar tráfico UDP en el Host 5 y el Host 18, medir delay, BW, Packet loss obtener una desviación estándar menor al 5%, cuando se envían 10Mb, 100Mb y 1000Mb de datos.  
R/

## **BIBLIOGRAFÍA**

- OpenFlow Message Layer: <http://flowgrammable.org/sdn/openflow/message-layer/>