

 DEPARTAMENTO DE INGENIERÍA TELEMÁTICA Y ELECTRÓNICA ETSIS TELECOMUNICACIÓN UPM	REDES Y SERVICIOS AVANZADOS Examen de convocatoria extraordinaria - Parte EC2. 30 de junio de 2022	
APELLOS:		
NOMBRE:		DNI:

- d) ¿Cuáles son los primeros mensajes Openflow que se intercambian el switch al que pertenece la tabla anterior y el controlador como consecuencia de la ejecución del ping anterior? Indique los mensajes Openflow que se observarán hasta que los dos flujos de la pregunta anterior están creados, y describa brevemente el propósito y el contenido fundamental de cada uno. No incluya en este análisis los mensajes ARP, es decir, puede suponer que las tablas ARP de los sistemas finales involucrados contienen la información necesaria. (1 punto)

- 1) **PACKET_IN:** Cuando el switch recibe el echo request por uno de sus puertos, utilizando la tercera entrada de la tabla de flujos enviará un **PACKET_IN** al controlador con dicho paquete.
- 2) **PACKET_OUT:** El controlador entregará a la aplicación Reactive Forwarding y esta, calculando el camino entre origen y destino, causará el envío de un **PACKET_OUT** que contiene la petición de eco al switch indicándole que lo envíe por el puerto que corresponda al camino calculado.
- 3) **FLOW_MOD:** La aplicación también causa que el controlador instale el flujo correspondiente al camino elegido entre origen y destino en la tabla de flujos (sería el flujo 5 o el 6, con los datos del enunciado no se puede saber si uno u otro).
- 4) **BARRIER_REQUEST:** Tras el **FLOW_MOD** el controlador se asegura de controlar el orden y el flujo de ejecución de las órdenes enviadas al switch mediante este mensaje, al que el switch contestará con un **BARRIER_REPLY**.
- 5) **PACKET_IN:** Cuando el switch recibe el echo reply por uno de sus puertos, lo envía al controlador por motivos análogos al paso 1)
- 6) **PACKET_OUT:** Lo envía el controlador por motivos análogos al paso 2) (para el tráfico "de vuelta").
- 7) **FLOW_MOD, BARRIER_REQUEST/REPLY:** Motivos análogos al paso 3) y 4), para el tráfico "de vuelta").

- c) Si un host conectado a SW1 intercambia tráfico del plano de datos con otro host conectado a SW3, ¿qué enlaces atravesará ese tráfico? Explique por qué, haciendo referencia específica a lo que ha rellenado en la figura como respuesta a la pregunta anterior. (0,2 puntos)

En el plano de datos, si un puerto está en estado blocking, el switch ni aprenderá direcciones MAC de tramas entrantes por dicho puerto, ni reenviará tráfico por el mismo, ni procesará tráfico entrante por ese puerto. Por tanto, cualquier enlace que tenga en uno de sus extremos un puerto en estado BLK, será un enlace no utilizable para el tráfico del plano de datos mientras el spanning tree no cambie.

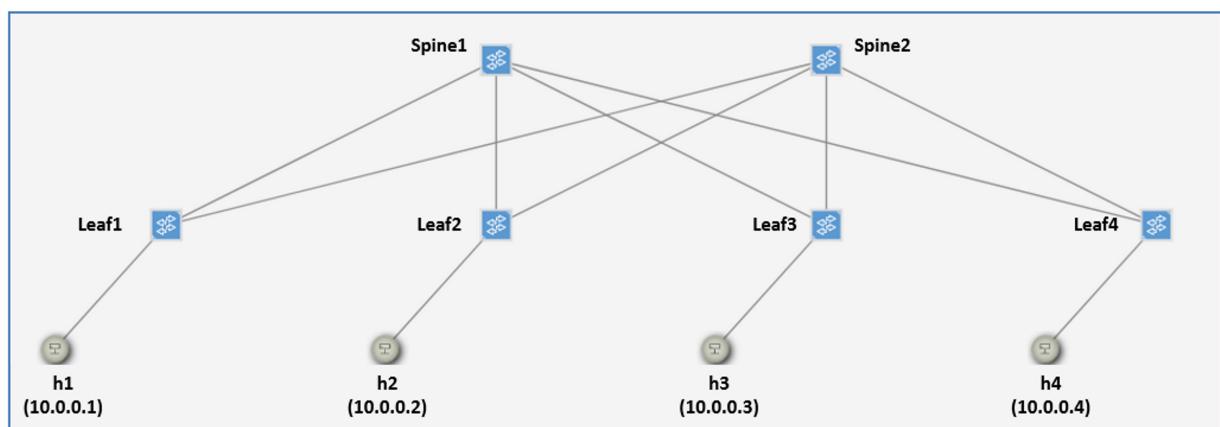
Por este motivo, el tráfico entre un host conectado a SW1 y un host conectado a SW3 necesariamente atravesará los siguientes enlaces:

Host conectado a SW1 - SW1 - SW2 - SW3 - host conectado a SW3

ya que, si se eliminan de la figura todos los enlaces no utilizables para el plano de datos, se trata del único camino para conectar SW1 y SW3.

EJERCICIO 8. Puntuación: 1 punto. Tiempo estimado: 15 minutos

La siguiente figura muestra el escenario de red emulada con Mininet utilizado en la Práctica 3 “Configuración de una red SDN y análisis de tráfico OpenFlow” de este curso, tal y como la muestra el controlador ONOS:



Como consecuencia de un ping realizado entre dos de los hosts se captura la siguiente trama por parte de Wireshark. Se trata de la primera trama que contiene un mensaje Openflow como consecuencia directa del ping mencionado, y en ella se han ocultado algunas líneas para facilitar la localización de la información:



APELLIDOS: **SOLUCIÓN**

NOMBRE: _____ DNI: _____

```

1 No.      Time        Source      Destination   Protocol    Length     Info
2 3542    29.236087954  172.17.0.1  172.17.0.2  OpenFlow    152       Type: OFPT_PACKET_IN
3
4 ▶ Frame 3542: 152 bytes on wire (1216 bits), 152 bytes captured (1216 bits) on interface 0
22 ◀ Linux cooked capture
23   Packet type: Sent by us (4)
24   Link-layer address type: 1
25   Link-layer address length: 6
26   Source: 02:42:f9:86:c2:46 (02:42:f9:86:c2:46)
27   Unused: 6400
28   Protocol: IPv4 (0x0800)
29 ▶ Internet Protocol Version 4, Src: 172.17.0.1, Dst: 172.17.0.2
48 ▶ Transmission Control Protocol, Src Port: 38608, Dst Port: 6653, Seq: 53487, Ack: 11509, Len: 84
93 ◀ OpenFlow 1.4
94   Version: 1.4 (0x05)
95   Type: OFPT_PACKET_IN (10)
96   Length: 84
97   Transaction ID: 0
98   Buffer ID: OFP_NO_BUFFER (4294967295)
99   Total length: 42
100  Reason: OFPR_APPLY_ACTION (1)
101  Table ID: 0
102  Cookie: 0x00010000e1a5f53e
103 ▶ Match
114 ◀ Data
115   Ethernet II, Src: e6:89:fe:a4:ab:68 (e6:89:fe:a4:ab:68), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
116   | Destination: Broadcast (ff:ff:ff:ff:ff:ff)
120   | Source: e6:89:fe:a4:ab:68 (e6:89:fe:a4:ab:68)
124 ◀   Type: ARP (0x0806)
125   Address Resolution Protocol (request)
126   | Hardware type: Ethernet (1)
127   | Protocol type: IPv4 (0x0800)
128   | Hardware size: 6
129   | Protocol size: 4
130   | Opcode: request (1)
131   | Sender MAC address: e6:89:fe:a4:ab:68 (e6:89:fe:a4:ab:68)
132   | Sender IP address: 10.0.0.3
133   | Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
134   | Target IP address: 10.0.0.4
135

```

El mensaje Openflow contenido en la trama anterior está directamente relacionado con una de las entradas de la siguiente tabla de flujos de uno de los nodos de la red:

STATE	PACKETS	DURATION	FLOW PRIORITY	TABLE NAME	SELECTOR	TREATMENT	APP NAME
Added	5	2,044	40000	0	ETH_TYPE:arp	imm[OUTPUT:CONTROLLER], cleared:true	*core
Added	12	2,044	5	0	ETH_TYPE:ipv4	imm[OUTPUT:CONTROLLER], cleared:true	*core
Added	1,319	2,044	40000	0	ETH_TYPE:bddp	imm[OUTPUT:CONTROLLER], cleared:true	*core
Added	1,319	2,044	40000	0	ETH_TYPE:lldp	imm[OUTPUT:CONTROLLER], cleared:true	*core

Con toda esta información, conteste a las siguientes cuestiones. Para todas ellas es imprescindible un razonamiento que haga referencia exacta a la información concreta contenida en la figura, la trama (incluyendo el número de línea cuando sea conveniente) y la tabla de flujos anterior. No son válidos razonamientos únicamente basados en lo que recuerda de lo sucedido en los experimentos realizados en la práctica si no los relaciona con la información contenida en este enunciado.

- a) ¿De qué host a qué host se ha realizado el ping? Razone la respuesta. (0,2 puntos)

De h3 a h4.

Razonamiento: la primera trama Openflow capturada es un PACKET_IN que lleva en el campo Data una Ethernet con una petición ARP encapsulada (la petición ARP se encuentra en las líneas 125 a 134 de la captura). Esta petición la está enviando el host con dirección IP 10.0.0.3 (línea 132) y pregunta por la dirección MAC que corresponde al host con dirección IP 10.0.0.4 (línea 134), y por tanto es la trama ARP generada como consecuencia de hacer un ping desde h3 hacia h4 (viendo la correspondencia entre host y direcciones IP de la figura que muestra la red).

- b) ¿Como consecuencia de cuál de los flujos de la tabla mostrada anteriormente se ha generado el paquete Openflow contenido en la captura? Razone la respuesta. (0,2 puntos)

El flujo que se muestra en primer lugar.

Razonamiento: el mensaje Openflow capturado es un PACKET_IN que en el campo Data contiene una trama Ethernet con un mensaje ARP encapsulado (líneas 115 a 134 de la captura). El único selector de la tabla que encaja con este paquete es el del primer flujo (selector ETH_TYPE:arp, es decir, una trama Ethernet que lleve encapsulado un paquete ARP), y las instrucciones (Treatment) a aplicar coinciden con la generación del PACKET_IN puesto que contienen OUTPUT:CONTROLLER, es decir, envío al controlador mediante un mensaje PACKET_IN.

- c) ¿A qué nodo pertenece la tabla de flujos? Razone la respuesta. (0,2 puntos)

Leaf3.

Razonamiento: Es el nodo que ha enviado el primer PACKET_IN al controlador con la petición de ARP, con lo que debe ser el nodo al que está conectado h3: Leaf3.

- d) ¿Qué contiene el campo Data del mensaje Openflow capturado? ¿De dónde obtiene esa información el nodo que genera ese mensaje? Razone la respuesta. (0,2 puntos)

Contiene la trama Ethernet que lleva encapsulada la petición de ARP generada por h3, tal y como le llegó a Leaf3 por el puerto que lo une a h3. Aplicando el tratamiento del primer flujo de la tabla de flujos mostrada anteriormente, Leaf3 incluye dicha trama Ethernet en un mensaje PACKET_IN de Openflow y se lo envía al controlador.

- e) ¿Cuál es el origen y el destino del mensaje Openflow contenido en la trama capturada? (no se piden las direcciones IP de origen y destino, sino los nodos o elementos del escenario de la práctica; debe identificarlos de manera precisa, ej. "h2" o "Spine1"). Razone la respuesta. (0,2 puntos)

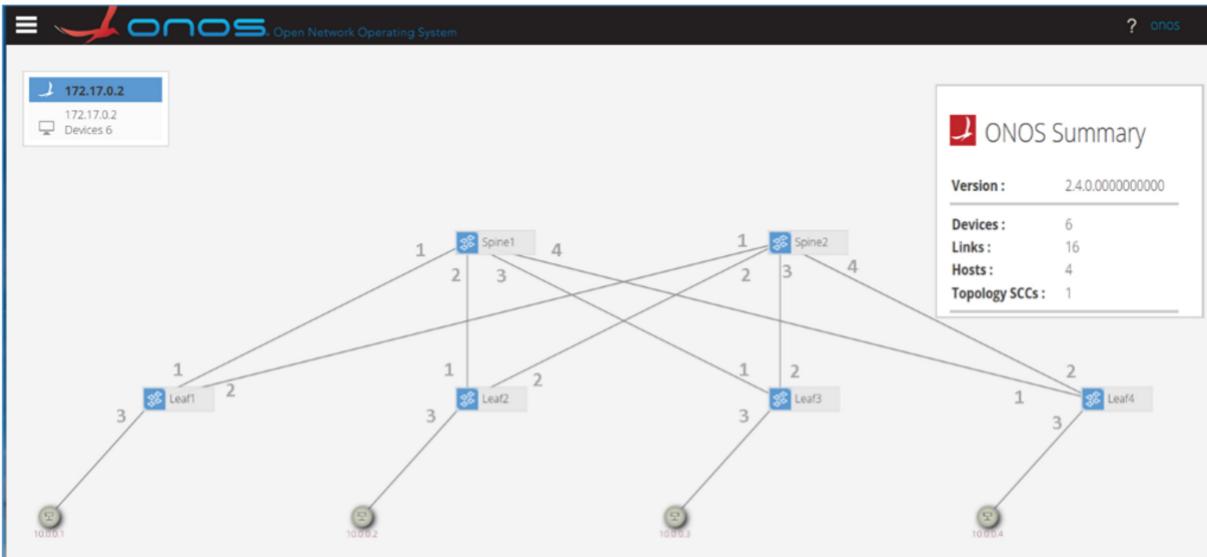
Origen: Leaf3. Destino: controlador ONOS.

Razonamiento: un mensaje Openflow de tipo PACKET_IN se envía de un switch a un controlador para hacerle llegar, en el campo Data, una trama recibida por uno de sus puertos y cuyo tratamiento a aplicar sea ese: enviarla al controlador. En la respuesta a la pregunta d) ya se indicó que este mensaje Openflow lo genera Leaf3, por lo que es el origen, siendo el destino el controlador ONOS, que al recibirlo tomará las acciones necesarias y contactará con la aplicación encargada de ese tipo de tráfico si es necesario.

 DEPARTAMENTO DE INGENIERÍA TELEMÁTICA Y ELECTRÓNICA ETSIS TELECOMUNICACIÓN UPM	REDES Y SERVICIOS AVANZADOS Examen de evaluación continua EC2. 28 de mayo de 2021	
APELLOS: SOLUCIÓN		
NOMBRE:	DNI:	

EJERCICIO 9. Puntuación: 1,5 puntos. Tiempo estimado: 15 minutos

En la figura siguiente se muestra la topología de la red “Spine&Leaf” que se utilizará para la resolución de este ejercicio sobre la práctica 3 “Configuración de una red SDN y análisis de tráfico OpenFlow”.



En las figuras siguientes se muestra la información proporcionada por el controlador ONOS sobre dispositivos de red, hosts y aplicaciones activadas.

Devices (6 total)								
FRIENDLY NAME	DEVICE ID	MASTER	PORTS	VENDOR	H/W VERSION	S/W VERSION	PROTOCOL	
✓	Leaf1	of:0000000000000003	172.17.0.2	4	Nicira, Inc.	Open vSwitch	2.9.8	OF_14
✓	Leaf2	of:0000000000000004	172.17.0.2	4	Nicira, Inc.	Open vSwitch	2.9.8	OF_14
✓	Leaf3	of:0000000000000005	172.17.0.2	4	Nicira, Inc.	Open vSwitch	2.9.8	OF_14
✓	Leaf4	of:0000000000000006	172.17.0.2	4	Nicira, Inc.	Open vSwitch	2.9.8	OF_14
✓	Spine1	of:0000000000000001	172.17.0.2	5	Nicira, Inc.	Open vSwitch	2.9.8	OF_14
✓	Spine2	of:0000000000000002	172.17.0.2	5	Nicira, Inc.	Open vSwitch	2.9.8	OF_14

Hosts (4 total)						
FRIENDLY NAME	HOST ID	MAC ADDRESS	VLAN ID	CONFIGURED	IP ADDRESSES	LOCATION
10.0.0.4	42:62:B4:4A:A6:D7/None	42:62:B4:4A:A6:D7	None	false	10.0.0.4	of:0000000000000006/3
10.0.0.3	12:84:D5:2B:49:64/None	12:84:D5:2B:49:64	None	false	10.0.0.3	of:0000000000000005/3
10.0.0.2	06:23:7B:66:D3:7E/None	06:23:7B:66:D3:7E	None	false	10.0.0.2	of:0000000000000004/3
10.0.0.1	D2:33:BC:A5:A0:FA/None	D2:33:BC:A5:A0:FA	None	false	10.0.0.1	of:0000000000000003/3

```

Terminal
File Edit View Search Terminal Help
Documentation: wiki.onosproject.org
Tutorials: tutorials.onosproject.org
Mailing lists: lists.onosproject.org

Come help out! Find out how at: contribute.onosproject.org

Hit '<tab>' for a list of available commands
and '[cmd] --help' for help on a specific command.
Hit '<ctrl-d>' or type 'logout' to exit ONOS session.

karaf@root > onos
karaf@root onos> apps -a -s
* 17 org.onosproject.drivers          2.4.0.SNAPSHOT Default Drivers
* 18 org.onosproject.optical-model    2.4.0.SNAPSHOT Optical Network Model
* 55 org.onosproject.openflow-base    2.4.0.SNAPSHOT OpenFlow Base Provider
* 56 org.onosproject.lldpprovider    2.4.0.SNAPSHOT LLDP Link Provider
* 57 org.onosproject.hostprovider    2.4.0.SNAPSHOT Host Location Provider
* 73 org.onosproject.openflow         2.4.0.SNAPSHOT OpenFlow Provider Suit
e
* 108 org.onosproject.gui2           2.4.0.SNAPSHOT ONOS GUI2
* 112 org.onosproject.layout          2.4.0.SNAPSHOT UI Auto-Layout
* 151 org.onosproject.proxyarp        2.4.0.SNAPSHOT Proxy ARP/NDP
karaf@root onos>

```

Teniendo en cuenta la información anterior, se pide:

- Indique razonadamente las configuraciones que deben realizarse en el controlador de ONOS para obtener el resultado mostrado en la siguiente figura con la ejecución del comando “pingall” en Mininet. (0,3 puntos):

```

Terminal
File Edit View Search Terminal Help
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 X X
h2 -> h1 X X
h3 -> X X h4
h4 -> X X h3
*** Results: 66% dropped (4/12 received)
mininet>

```

Analizando el resultado del comando “pingall” de Mininet, se ve que hay conectividad entre “h1-h2” y entre “h3-h4”.

Además, en la lista de aplicaciones activadas mediante el comando “apps -a -s” se observa que el controlador no tiene activada la aplicación de “Reactive Forwarding” (org.onosproject.fwd).

Por lo tanto, se tienen configurados dos “host-to-host intent”, uno entre h1 y h3 y otro entre h2 y h4 para permitir el tráfico de datos entre dichos hosts.

 DEPARTAMENTO DE INGENIERÍA TELEMÁTICA Y ELECTRÓNICA ETSIS TELECOMUNICACIÓN UPM	REDES Y SERVICIOS AVANZADOS Examen de evaluación continua EC2. 28 de mayo de 2021	
APELLOS: SOLUCIÓN		
NOMBRE:	DNI:	

- b) Indique al menos dos formas diferentes de realizar las configuraciones del apartado a) sin utilizar REST API. No es necesario especificar los comandos concretos para la configuración. (0,2 puntos)

Se puede hacer desde la interfaz de comandos (CLI) del controlador, mediante el comando "add-host-intent" y desde la interfaz gráfica (GUI) del controlador, seleccionando la pareja de host de cada "intent".

- c) Suponiendo que el dispositivo Spine2 se ha desconectado por avería, escriba el contenido de la tabla de flujos en el dispositivo Spine1. Rellene únicamente las filas necesarias para los flujos creados al realizar la configuración del apartado a). (0,6 puntos)

FLows	SELECTOR	TREATMENT
1	IN_PORT: 1 , ETH_DST: 06:23:7B:66:D3:7E , ETH_SRC: D2:33:BC:A5:A0:FA	imm[OUTPUT: 2], cleared: false
2	IN_PORT: 2 , ETH_DST: D2:33:BC:A5:A0:FA , ETH_SRC: 06:23:7B:66:D3:7E	imm[OUTPUT: 1], cleared: false
3	IN_PORT: 3 , ETH_DST: 42:62:B4:4A:A6:D7 , ETH_SRC: 12:84:D5:2B:49:64	imm[OUTPUT: 4], cleared: false
4	IN_PORT: 4 , ETH_DST: 12:84:D5:2B:49:64 , ETH_SRC: 42:62:B4:4A:A6:D7	imm[OUTPUT: 3], cleared: false
5		
6		

- d) Rellene los parámetros siguientes con los valores adecuados para la configuración del apartado a) utilizando REST API y JSON como formato de intercambio de datos entre la aplicación y el controlador ONOS. Repita los parámetros de codificación en JSON tantas veces como sea necesario. (0,4 puntos)

Categoría comando ONOS: **Intent**

Método REST-API: **POST**

Codificación en JSON:

```
{  
    "type": "HostToHostIntent",  
    "appId": "org.onosproject.ovsdb",  
    "priority": 55,  
    "one": "D2:33:BC:A5:A0:FA/-1",  
    "two": "06:23:7B:66:D3:7E/-1"  
}  
  
{  
    "type": "HostToHostIntent",  
    "appId": "org.onosproject.ovsdb",  
    "priority": 55,  
    "one": "12:84:D5:2B:49:64/-1",  
    "two": "42:62:B4:4A:A6:D7/-1"  
}
```

APELLIDOS: **SOLUCIÓN**

NOMBRE:

DNI:

EJERCICIO 6. Puntuación: 2,5 puntos. Tiempo estimado: 20 minutos

Las preguntas de este ejercicio están relacionadas con lo aprendido y ejercitado en la práctica 3 (*Configuración de una red SDN y análisis de tráfico OpenFlow*) de la asignatura.

- a) Describa brevemente la funcionalidad de los mensajes OFPT_BARRIER_REQUEST/REPLY (entre qué elementos se intercambia cada uno, en qué sentido, qué significado y propósito tiene cada uno). (0,5 puntos)

OFPT_BARRIER_REQUEST: De controlador a switch. Permite al controlador tener control sobre el orden en que se procesan conjuntos de mensajes Openflow por parte del switch, ya que el switch que recibe este mensaje debe finalizar todos los mensajes recibidos previamente en esa conexión (incluyendo el envío de las correspondientes respuestas) antes de ejecutar el BARRIER_REQUEST o cualquier mensaje posterior a él.

OFPT_BARRIER_REPLY: De switch a controlador, en respuesta al correspondiente REQUEST, y solo cuando ya se ha cumplido la condición explicada en el párrafo anterior

En la práctica, estando activada la aplicación “Reactive forwarding”, al ejecutar la orden “`h1 ping -c1 h3`” en la consola de Mininet, se observa tráfico ICMP y Openflow en las capturas realizadas.

Se muestran a continuación capturas de tráfico Openflow intercambiadas entre el controlador ONOS y uno de los switches involucrados en el tráfico ICMP relativo al ping, con distinto nivel de detalle.

Flujo general de paquetes Openflow entre controlador y switch como consecuencia del ping:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	172.17.0.1	172.17.0.2	OpenFlow	208	Type: OFPT_PACKET_IN
2	0.007083800	172.17.0.2	172.17.0.1	OpenFlow	206	Type: OFPT_PACKET_OUT
3	0.025244878	172.17.0.2	172.17.0.1	OpenFlow	180	Type: OFPT_BARRIER_REQUEST
4	0.025440909	172.17.0.1	172.17.0.2	OpenFlow	76	Type: OFPT_BARRIER_REPLY
5	0.035033847	172.17.0.1	172.17.0.2	OpenFlow	208	Type: OFPT_PACKET_IN
6	0.035729175	172.17.0.2	172.17.0.1	OpenFlow	206	Type: OFPT_PACKET_OUT
7	0.036226171	172.17.0.2	172.17.0.1	OpenFlow	180	Type: OFPT_BARRIER_REQUEST
8	0.036315850	172.17.0.1	172.17.0.2	OpenFlow	76	Type: OFPT_BARRIER_REPLY

Detalle de la trama No. 1:

```
1 Frame 1: 208 bytes on wire (1664 bits), 208 bytes captured (1664 bits) on interface any, id 0
2 Linux cooked capture
3 Internet Protocol Version 4, Src: 172.17.0.1, Dst: 172.17.0.2
4 Transmission Control Protocol, Src Port: 34578, Dst Port: 6653, Seq: 1, Ack: 1, Len: 140
5 OpenFlow 1.4
6   Version: 1.4 (0x05)
7   Type: OFPT_PACKET_IN (10)
8   Length: 140
9   Transaction ID: 0
10  Buffer ID: OFP_NO_BUFFER (4294967295)
11  Total length: 98
12  Reason: OFPR_APPLY_ACTION (1)
13  Table ID: 0
14  Cookie: 0x00010000be641e06
15  Match
16    Type: OFPPMT_OXM (1)
17    Length: 12
18    OXM field
19      Class: OFPXMC_OPENFLOW_BASIC (0x8000)
20      0000 000. = Field: OFPXMT_OFB_IN_PORT (0)
21      .... ...0 = Has mask: False
22      Length: 4
23      Value: 3
24      Pad: 00000000
25      Pad: 0000
26  Data
27    Ethernet II, Src: 1a:a8:e4:4f:e8:59 (1a:a8:e4:4f:e8:59), Dst: 06:0b:d3:15:52:d2 (06:0b:d3:15:52:d2)
28    Internet Protocol Version 4, Src: 10.0.0.1, Dst: 10.0.0.3
29    Internet Control Message Protocol
30      Type: 8 (Echo (ping) request)
31      Code: 0
32      Checksum: 0x9bf6 [correct]
33      [Checksum Status: Good]
34      Identifier (BE): 11085 (0x2b4d)
35      Identifier (LE): 19755 (0x4d2b)
36      Sequence number (BE): 1 (0x0001)
37      Sequence number (LE): 256 (0x0100)
38      [No response seen]
39      Timestamp from icmp data: Apr 14, 2021 23:31:04.000000000 Hora de verano romance
40      [Timestamp from icmp data (relative): 0.404054841 seconds]
41      Data (48 bytes)
```

Detalle de la trama No. 2:

```
1 Frame 2: 206 bytes on wire (1648 bits), 206 bytes captured (1648 bits) on interface any, id 0
2 Linux cooked capture
3 Internet Protocol Version 4, Src: 172.17.0.2, Dst: 172.17.0.1
4 Transmission Control Protocol, Src Port: 6653, Dst Port: 34578, Seq: 1, Ack: 141, Len: 138
5 OpenFlow 1.4
6   Version: 1.4 (0x05)
7   Type: OFPT_PACKET_OUT (13)
8   Length: 138
9   Transaction ID: 0
10  Buffer ID: OFP_NO_BUFFER (4294967295)
11  In port: 3
12  Actions length: 16
13  Pad: 000000000000
14  Action
15    Type: OFPAT_OUTPUT (0)
16    Length: 16
17    Port: 2
18    Max length: 0
19    Pad: 000000000000
20  Data
21    Ethernet II, Src: 1a:a8:e4:4f:e8:59 (1a:a8:e4:4f:e8:59), Dst: 06:0b:d3:15:52:d2 (06:0b:d3:15:52:d2)
22    Internet Protocol Version 4, Src: 10.0.0.1, Dst: 10.0.0.3
23    Internet Control Message Protocol
24      Type: 8 (Echo (ping) request)
25      Code: 0
26      Checksum: 0x9bf6 [correct]
27      [Checksum Status: Good]
28      Identifier (BE): 11085 (0x2b4d)
29      Identifier (LE): 19755 (0x4d2b)
30      Sequence number (BE): 1 (0x0001)
31      Sequence number (LE): 256 (0x0100)
32      [Response frame: 5]
33      Timestamp from icmp data: Apr 14, 2021 23:31:04.000000000 Hora de verano romance
34      [Timestamp from icmp data (relative): 0.411138641 seconds]
35      Data (48 bytes)
```

APELLIDOS: **SOLUCIÓN**

NOMBRE:

DNI:

Detalle de la trama No. 3:

```

1 Internet Protocol Version 4, Src: 172.17.0.2, Dst: 172.17.0.1
2 Transmission Control Protocol, Src Port: 6653, Dst Port: 34578, Seq: 139, Ack: 141, Len: 112
3 OpenFlow 1.4
4     Version: 1.4 (0x05)
5     Type: OFPT_FLOW_MOD (14)
6     Length: 104
7     Transaction ID: 72
8     Cookie: 0x00510000cde527d3
9     Cookie mask: 0x0000000000000000
10    Table ID: 0
11    Command: OFPFC_ADD (0)
12    (...)

13    Match
14        Type: OFPMT_OXM (1)
15        Length: 32
16        OXM field
17            Class: OFPXMC_OPENFLOW_BASIC (0x8000)
18            0000 000. = Field: OFPXMT_OFB_IN_PORT (0)
19            .... ...0 = Has mask: False
20            Length: 4
21            Value: 3
22        OXM field
23            Class: OFPXMC_OPENFLOW_BASIC (0x8000)
24            0000 011. = Field: OFPXMT_OFB_ETH_DST (3)
25            .... ...0 = Has mask: False
26            Length: 6
27            Value: 06:0b:d3:15:52:d2 (06:0b:d3:15:52:d2)
28        OXM field
29            Class: OFPXMC_OPENFLOW_BASIC (0x8000)
30            0000 100. = Field: OFPXMT_OFB_ETH_SRC (4)
31            .... ...0 = Has mask: False
32            Length: 6
33            Value: 1a:a8:e4:4f:e8:59 (1a:a8:e4:4f:e8:59)
34    Instruction
35        Type: OFPIT_APPLY_ACTIONS (4)
36        Length: 24
37        Pad: 00000000
38        Action
39            Type: OFPAT_OUTPUT (0)
40            Length: 16
41            Port: 2
42            Max length: 0
43            Pad: 000000000000
44    OpenFlow 1.4
45        Version: 1.4 (0x05)
46        Type: OFPT_BARRIER_REQUEST (20)
47        Length: 8
48        Transaction ID: 72

```

Conteste a las siguientes cuestiones de manera razonada, precisa y haciendo referencia a las líneas de las capturas mostradas anteriormente. Incluya todos los detalles relacionados con lo que se pregunta (números de puerto, direcciones IP o MAC, ...) que pueda inferir de las capturas.

- b) Describa cuál ha sido el flujo de acontecimientos que ha llevado al nodo que ha generado la trama 1 a enviarla. (0,5 puntos)

Es un PACKET_IN (línea 7), por lo que se envía desde el switch al controlador como consecuencia de haber recibido una trama en uno de sus puertos de entrada (en concreto en el puerto 3, línea 23) y que la entrada de la tabla de flujos correspondiente indicó que debía enviarse al controlador (línea 12). En concreto, la trama recibida contiene una solicitud de eco de ICMP (líneas 27 a 41), que es la petición relacionada con el ping entre h1 y h3, cuyas direcciones MAC e IP se pueden observar en las líneas 27 y 28.

- c) Describa quién envía a quién la trama 2, y qué propósito tiene esta trama. (0,5 puntos)

Es un PACKET_OUT (línea 7), por lo que se envía del controlador al switch contenido un paquete en su campo Data e indicando la acción a realizar con él por parte del switch. En este caso la trama que se envía al switch es el mismo Echo request indicado en la respuesta al apartado anterior (líneas 21 a 35) y lo que se le dice al switch es que lo envíe (línea 15) por el puerto 2 (línea 17).

- d) Describa quién envía a quién la trama 3 y qué contiene. Es necesario que incluya esta información: qué función tiene(n) el o los mensajes contenidos en la misma, explicar los valores más importantes del campo "Match", y explicar los valores más importantes del campo "Instruction", haciendo referencia a valores observados en las capturas de las tramas 1 y 2 cuando sea pertinente. (1 punto)

La trama 3 contiene dos mensajes Openflow enviados del controlador al switch:

Mensaje FLOW_MOD (línea 5)

Ordena añadir un flujo (línea 11) a la tabla 0 (línea 10) de flujos del switch. Esta entrada tendrá los siguientes campos:

- Match: especifica que la trama cumpla estas condiciones:
 - Llegue por el puerto de entrada 3 (líneas 18 y 21).
 - Tenga como dirección MAC de destino (línea 24) la contenida en la línea 27.
 - Tenga como dirección MAC de origen (línea 30) la contenida en la línea 33.

Nota: son los parámetros asociados al Echo request que el switch envió al controlador en la trama 1.

- Instrucciones a realizar cuando una trama coincida con ese "Match"(campo "Instruction"):

- Enviar esa trama (línea 39) por el puerto 2.

Nota: es el puerto por el que se ordenó enviar el Echo request al switch mediante la trama 2.

Mensaje BARRIER_REQUEST (línea 46):

Ordena al switch que todos los mensajes Openflow previos de esta conexión se ejecuten y una vez hecho se responda con BARRIER_REPLY.