1. 測試內容:

測試Flow之match field: “SCTP\_SRC”, “SCTP\_DST” 是否正確運行。

1. 測試環境架構:



1. 測試環境組態

* SDN controller: ONOS 1.15.0
* Switch: DGS-3630
* Hosts:
* ubuntu 16.04.5 LTS
* dlinktest1(實體機)
  + - IP : 192.168.202.1
    - MAC : d0:94:66:5d:5e:08
    - Port : 1
* dlinktest2(實體機)
  + - IP : 192.168.202.2
    - MAC : d0:94:66:5c:45:20
    - Port : 2
* Apps:



1. 使用之JSON文件

* Flow rule: (附檔: SCTP.json)

IP\_PROTO == 132為SCTP\_SRC與SCTP\_DST的先備條件。ETH\_TYPE == 0x0800為IP\_PROTO的先備條件。

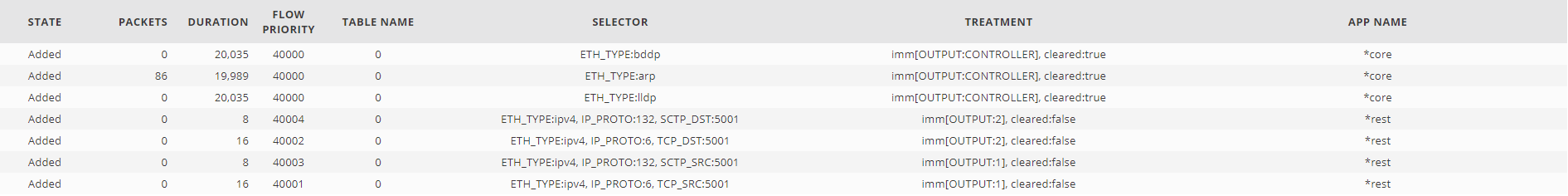
|  |
| --- |
| {  "flows":[  {  "priority":40001,  "timeout":0,  "isPermanent":true,  "deviceId":"of:000078321bdf4000",  "selector":{  "criteria":[  {  "type":"ETH\_TYPE",  "ethType":"0x0800"  },  {  "type":"IP\_PROTO",  "protocol":132  },  {  "type":"SCTP\_SRC",  "sctpPort":5001  }  ]  },  "treatment":{  "instructions":[  {  "type":"OUTPUT",  "port":"1"  }  ]  }  },  {  "priority":40001,  "timeout":0,  "isPermanent":true,  "deviceId":"of:000078321bdf4000",  "selector":{  "criteria":[  {  "type":"ETH\_TYPE",  "ethType":"0x0800"  },  {  "type":"IP\_PROTO",  "protocol":132  },  {  "type":"SCTP\_DST",  "sctpPort":5001  }  ]  },  "treatment":{  "instructions":[  {  "type":"OUTPUT",  "port":"2"  }  ]  }  }  ]  } |

* Flow rule (for TCP): (附檔: TCP.json)
* IP\_PROTO == 6為TCP\_SRC與TCP\_DST的先備條件。ETH\_TYPE == 0x0800為IP\_PROTO的先備條件。
* 需要針對TCP做處理是因為iperf3 client在啟動時需要透過TCP連接到server。

|  |
| --- |
| {  "flows":[  {  "priority":40001,  "timeout":0,  "isPermanent":true,  "deviceId":"of:000078321bdf4000",  "selector":{  "criteria":[  {  "type":"ETH\_TYPE",  "ethType":"0x0800"  },  {  "type":"IP\_PROTO",  "protocol":6  },  {  "type":"TCP\_SRC",  "tcpPort":5001  }  ]  },  "treatment":{  "instructions":[  {  "type":"OUTPUT",  "port":"1"  }  ]  }  },  {  "priority":40001,  "timeout":0,  "isPermanent":true,  "deviceId":"of:000078321bdf4000",  "selector":{  "criteria":[  {  "type":"ETH\_TYPE",  "ethType":"0x0800"  },  {  "type":"IP\_PROTO",  "protocol":6  },  {  "type":"TCP\_DST",  "tcpPort":5001  }  ]  },  "treatment":{  "instructions":[  {  "type":"OUTPUT",  "port":"2"  }  ]  }  }  ]  } |

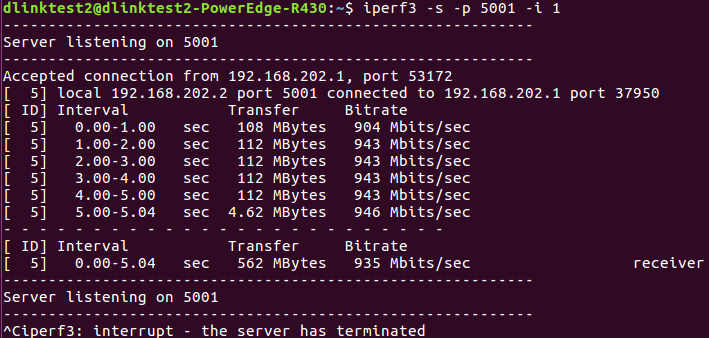
1. 測試
2. 測試一：SCTP\_SRC、SCTP\_DST match

* SCTP.json中的flow rules會將SCTP Destination port為5001的封包forward給h2；SCTP Source port為5001的封包forward給h1。因此若h1發送SCTP封包給listen在port 5001的h2，預期行為會是兩者可以互通。
* ONOS GUI: Flow View



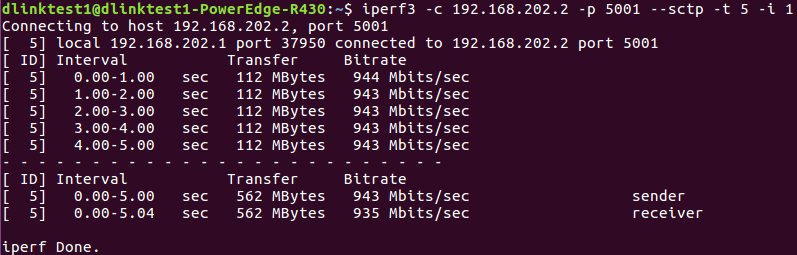
* h2 作為iperf3 server listen在port 5001。

|  |
| --- |
| iperf3 –s –p 5001 –i 1 |



* h1作為iperf3 client向h2發送SCTP封包。

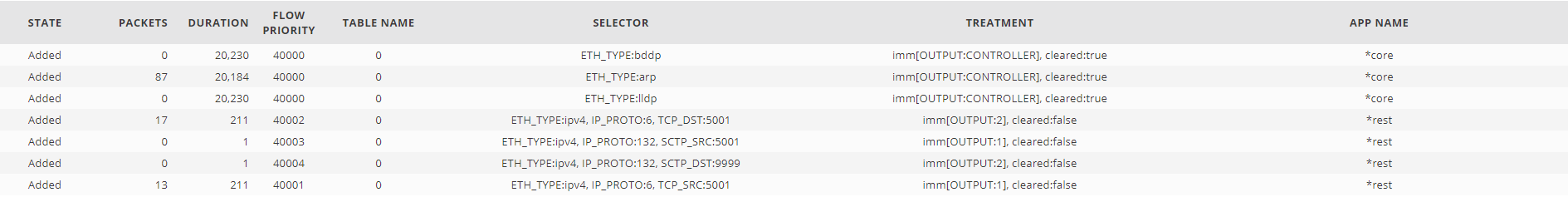
|  |
| --- |
| iperf3 –c 192.168.202.2 –p 5001 –sctp –t 5 –i 1 |



* 可以發現雙方確實能夠互通。

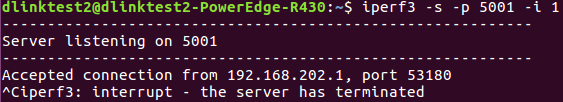
1. 測試二：SCTP\_DST not match

* 單獨將SCTP.json中SCTP\_DST對應的sctpPort值改成9999。如此一來h1發送給h2 (依舊listen在port 5001) 的SCTP封包將無法送達。因此利用tcpdump monitor s1-eth2的預期行為是捕捉不到SCTP封包。
* ONOS GUI: Flow View



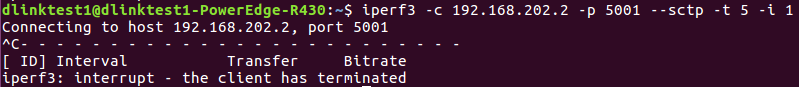
* h2 作為iperf3 server listen在port 5001。

|  |
| --- |
| iperf3 –s –p 5001 –i 1 |



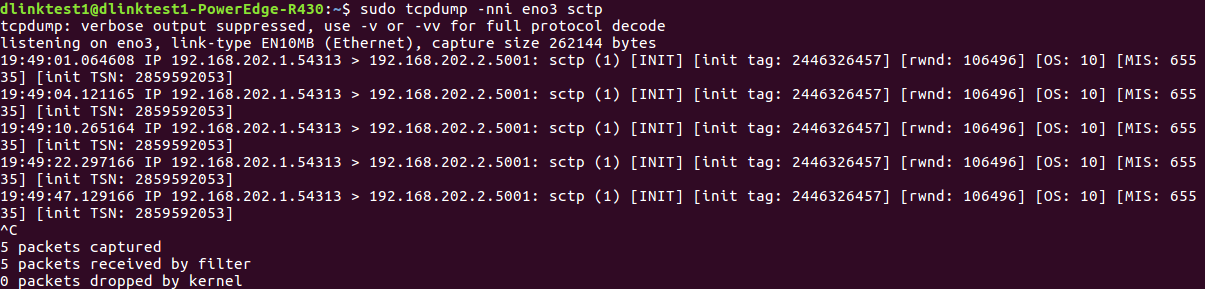
* h1作為iperf3 client向h2發送SCTP封包。

|  |
| --- |
| iperf3 –c 192.168.202.2 –p 5001 –sctp –t 5 –i 1 |



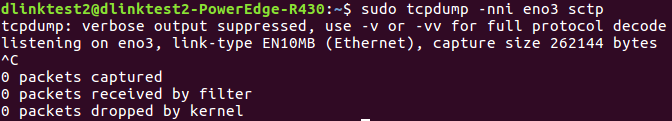
* 觀察 dlinktest1發現eno3一直重送SCTP INIT封包。

|  |
| --- |
| sudo tcpdump –nni eno3 sctp |



* 觀察dlinktest2發現eno3沒有收到SCTP INIT封包。，符合預期行為。

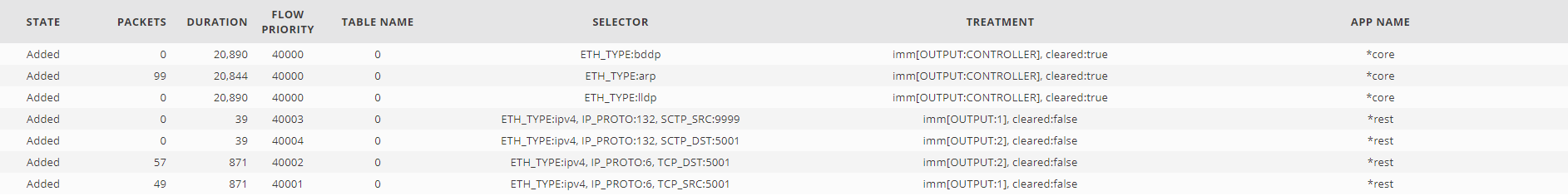
|  |
| --- |
| sudo tcpdump –nni eno3 sctp |



1. 測試三：SCTP\_SRC not match

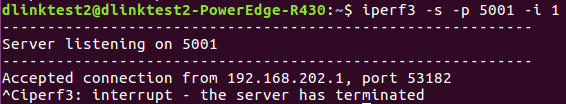
單獨將SCTP.json中SCTP\_SRC對應的sctpPort值改成9999。如此一來h2發送給h1的SCTP封包將無法送達。因此利用tcpdump monitor s1-eth1的預期行為是捕捉不到Source port 為5001的SCTP封包。

* ONOS GUI: Flow View



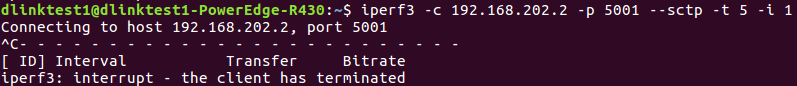
* h2 作為iperf3 server listen在port 5001。

|  |
| --- |
| iperf3 –s –p 5001 –i 1 |



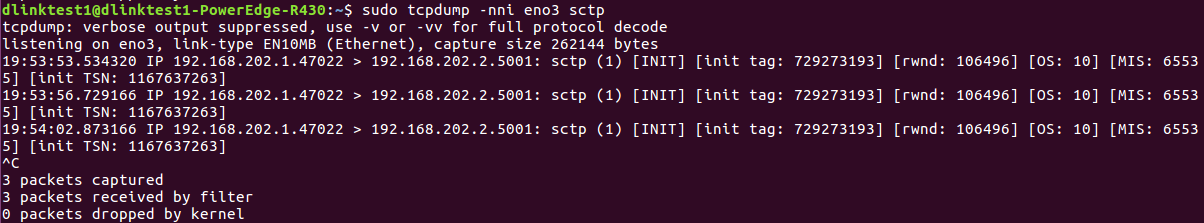
* h1作為iperf3 client向h2發送SCTP封包。

|  |
| --- |
| iperf3 –c 192.168.202.2 –p 5001 –sctp –t 5 –i 1 |



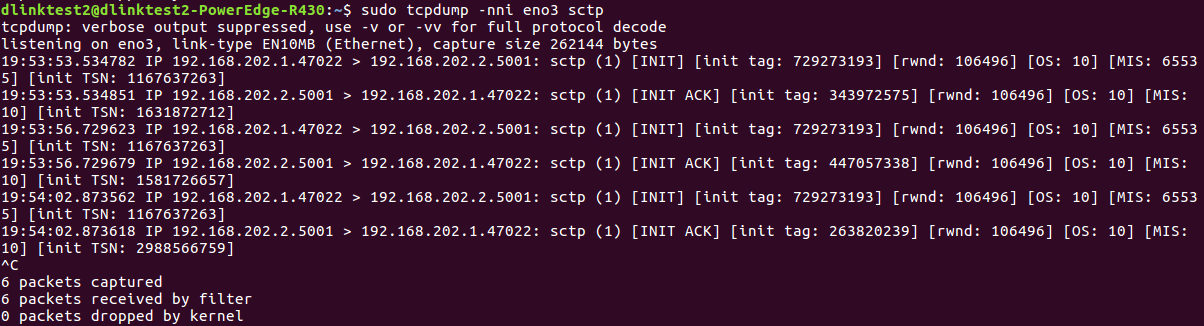
* 觀察 dlinktest1發現eno3一直重送SCTP INIT封包但沒有收到INIT ACK。

|  |
| --- |
| sudo tcpdump –nni eno3 sctp |



* 觀察dlinktest2發現eno3收到SCTP INIT封包且回傳INIT ACK，符合預期行為。

|  |
| --- |
| sudo tcpdump –nni eno3 sctp |



1. 測試結果

Flow之match field: “SCTP\_SRC”, “SCTP\_DST”正常運行。