Project 2 - OpenFlow Protocol Observation and Flow Rule Installation

112109045 何逸君

Part1

1 . How many OpenFlow headers with type “OFPT_FLOW_MOD” and command “OFPFC_ADD” are there among all the packets?

A: 10

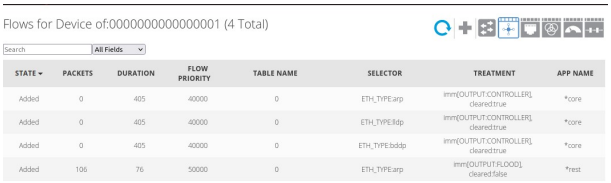
2 . What are the match fields and the corresponding actions in each “OFPT_FLOW_MOD” message?

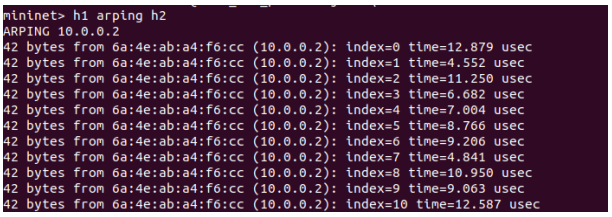
3 . What are the Idle Timeout values for all flow rules on s1 in GUI?

Match fields **Timeout Values**	**Action**	**Timeout Values**
ETH_TYPE = IPv4 (0x0800)	Output port = OFPP_CONTROLLER	0
ETH_TYPE = BDDP (0x8942)(Unknown)	Output port = OFPP_CONTROLLER	0
ETH_TYPE = APR (0x0806)	Output port = OFPP_CONTROLLER	0
ETH_TYPE = 802.1 Link Layer Discovery Protocol ,LLDP (0x88c)	Output port = OFPP_CONTROLLER	0
IN_PORT = 1	Output port = 2	0
ETH_DST = 72:66:09:c2:f1:ea	Output port = 2	0
ETH_SRC = 92:cc:64:fc:1a:27	Output port = 2	0
IN_PORT = 2	Output port = 1	0
ETH_DST = 92:cc:64:fc:1a:27	Output port = 1	0
ETH_SRC = 72:66:09:c2:f1:ea	Output port = 1	0

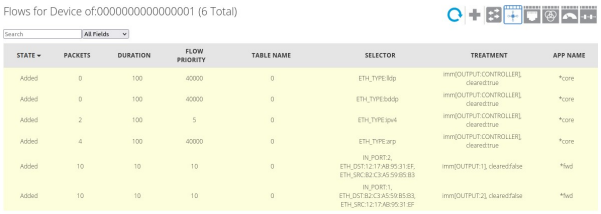
Part2

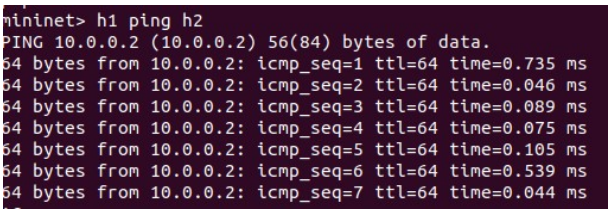
[one flow rule] Screenshot of the flow rules I installed:





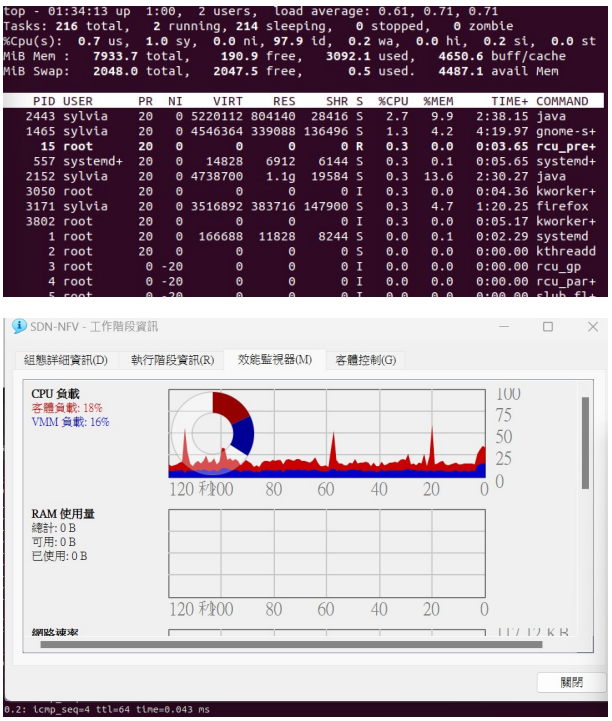
[two flow rules] Screenshot of the flow rules I installed:



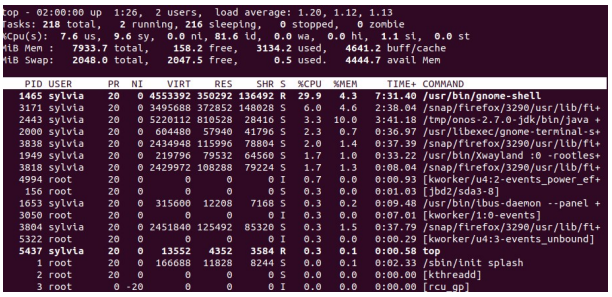


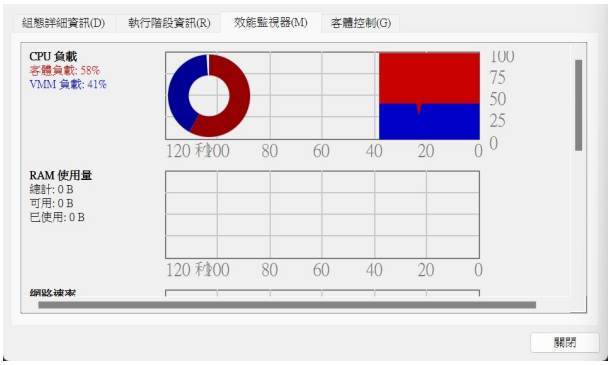
Part3

Screenshot of CPU's utilization - Before:



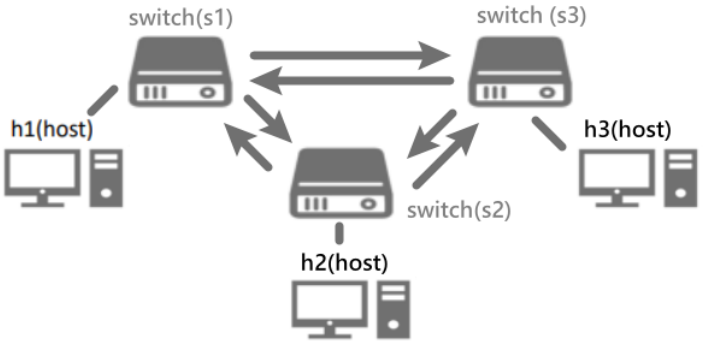
Screenshot of CPU's utilization - After: (top 中CPU的load average超過1，幾乎是原來的兩倍)





Q: Why the broadcast storm occur?

A: 我先創造出以下的topology，接著安裝相對應的flow rules，接著在mininet中執行h1 arping h2。從示意圖中可以看到h1會先送出一個arp broadcast封包，switch (s1) 收到從h1來的封包後會broadcast到switch (s2)與switch (s3)，switch (s2)收到從switch (s1)來的封包會broadcast到switch (s3)與h2，而switch (s3)收到封包後會在broadcast到switch (s2) 與 switch (s1)，如此不斷持續broadcast下去，最終就會導致broadcast storm.



Part4

Activate only “org.onosproject.fwd” and other initially activated APPs.

Use Mininet default topology and let h1 ping h2.

Q: What happens in control and data planes?

Data plane:

1. h1將ping packets forward到h2时，packets 先從h1 forward到 s1
2. s1收到packets 時，因為最初沒有適用的flow rule，因此將packets事件訊號先傳給controller
3. packets forward到controller，由controller做處理與轉發
4. 根據Reactive Forwarding app的ONOS controller產生的flow rule，controller將packets forward到相對應的switch以forward packets到目標host

Control plane:

1. Reactive Forwarding app的ONOS controller接收到packets事件訊號。controller會先讀取並檢查packets(ex: DST, SRC..etc)，並決定處理方式
2. controller根據packets內容，創建flow rule以將packets forward到h2，而後將其安裝到發送packets事件訊號的switch中，以flow rule命令switch根據符合目標特徵的packet進行什麼action
3. switch安裝好flow rule後，SDN switch會根據指令，將後續符合類似條件且來自h1的packets forward到h2。
4. 當h2收到ping的packets時，他會產生一個預計回覆訊號給h1
5. 當packets到達連接h2的switch時，switch會查看table找出匹配的flow rule，並根據其中內容，將回覆訊號回給h1

6. 當h1收到回覆，即會顯示ping連接成功

What I’ve learned or solved ?

- 熟悉**flow rule:** 透過part 1/2 兩小題練習flow rule的寫法以及須注意的事項。由於未注意到整體架構，先把mininet&flow rule重置才做part2第二小題，導致packets傳不出去，而更懂得ping連接的要素。
- 熟悉**wireshark:** 透過part 1/4 兩題練習解讀wireshark每個欄位的大致意義與架構。