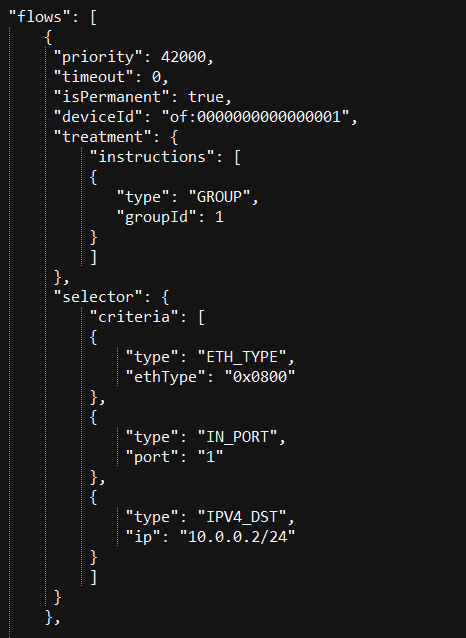
**REPORT**

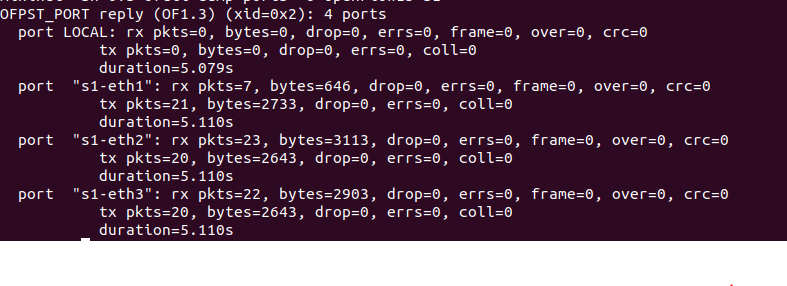
1. GROUP Type
2. 經過REST上傳group.json，group table的SELECT會經過hash，去選擇要執行哪個buckets的內容，換言之，雖然flow不會一次走port2、一次走port3，但經過幾次後，一定會走到另一個bucket的內容!



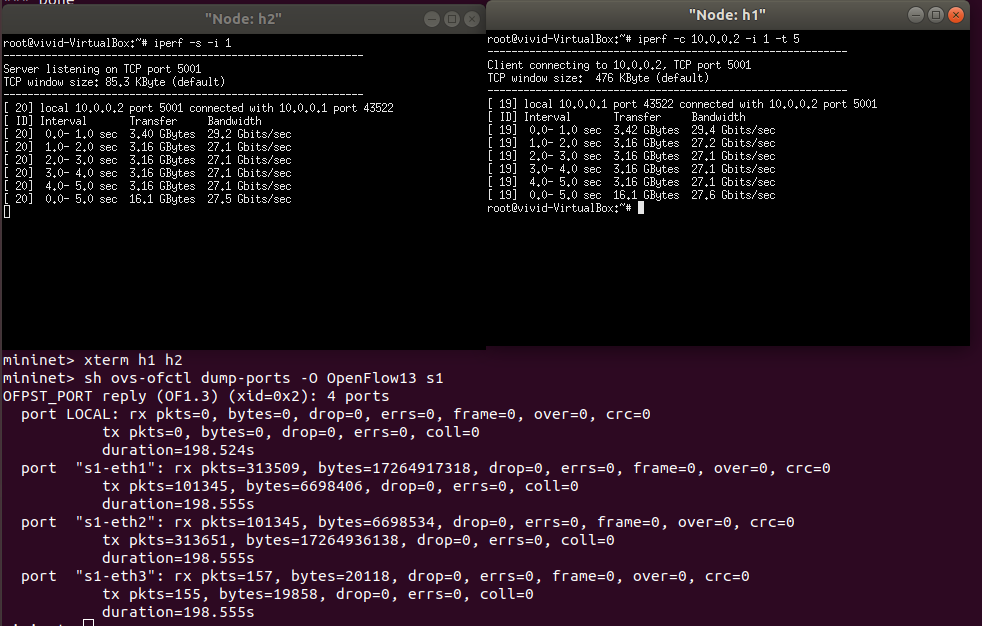
1. 經過REST 一次上傳多個flow rule，下圖為switch接收到來自h1->h2的flow，會將flow送往group



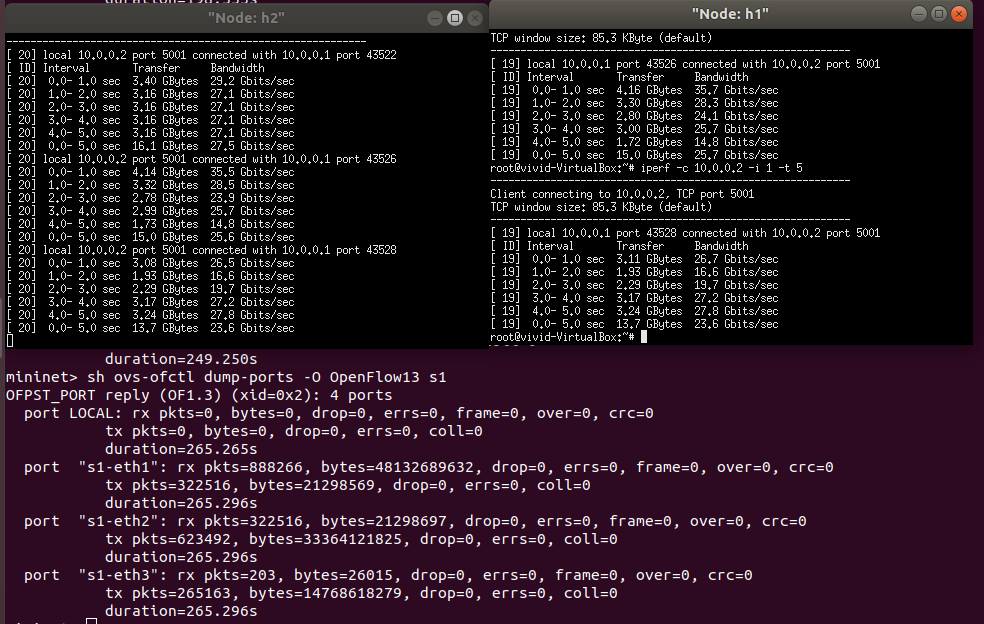
1. 接下來就可以到ONOS UI看FLOW有沒有成功，並且在Mininet裡輸入”xterm h1 h2”，就可以利用Iperf打流量，並且輸入” sh ovs-ofctl dump-ports -O OpenFlow13 s1” 利用switch1觀察h1->h2的路徑!
2. 還沒iperf前



1. Iperf後可以看到先走s1->s2的路徑



1. 再經過2次iperf後，看到flow走s1->s3的路徑



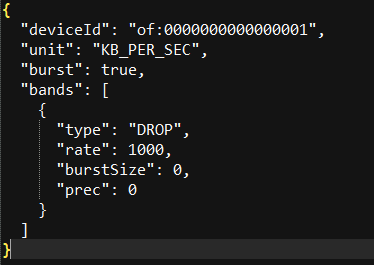
1. Meter Type

1. 建立sudo mn --controller=remote --topo single,2 的topology

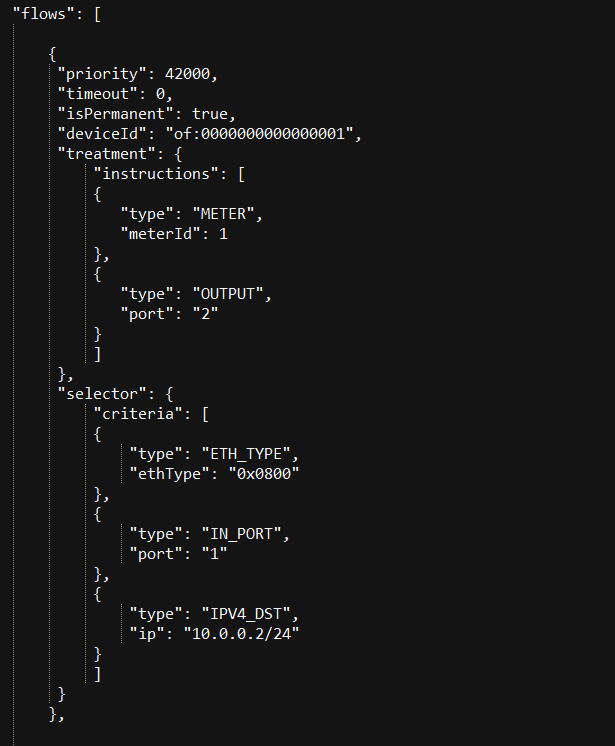
2. 輸入s1 ovs-vsctl set bridge "s1" datapath\_type=netdev，否則會有以下error.



3. 經過REST上傳meter.json，將TYPE設定為DROP，rate設定1000KB/s



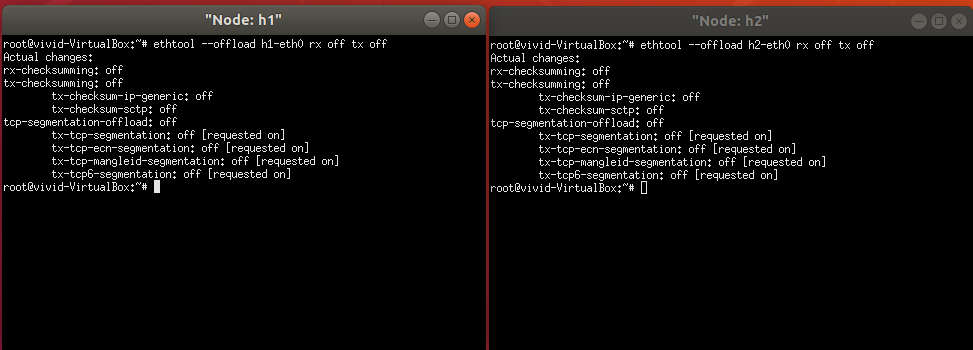
4. 經過REST上傳flow\_for\_meter.json



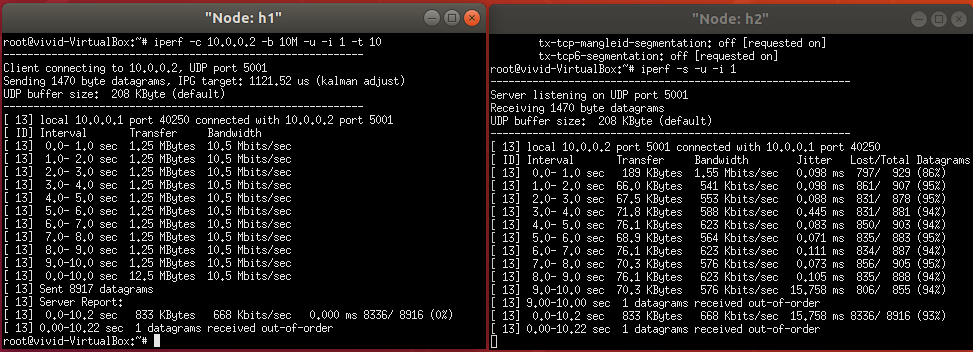
5. 設定完後同樣的開啟h1及h2，並且設定host的虛擬網卡不要進行rx 以及tx的checksum，如此一來才能使iperf正常運行!

**H1輸入: ethtool --offload h1-eth0 rx off tx off**

**H2輸入: ethtool --offload h2-eth0 rx off tx off**



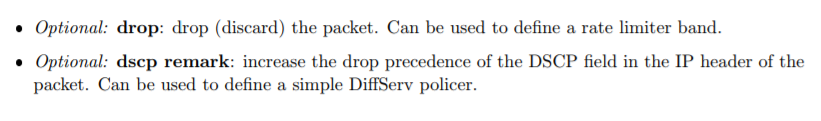
6. 利用iperf觀察流量變化，可以發現儘管h1送出10M/s的封包，h2卻只 收到的bandwidth低於1M/s的封包，有此可以見到Meter對traffic speed的限制!



1. Learn and solve:

1. 在看ONOS dovs裡的GROUP時，不需要有weight，所以一直沒辦法成功上傳group.json，之後到onos的github找才看到現在的版本是需要weight!

2. 在設定meter.json的時候要將band type從REMARK改成DROP才能夠發揮限速的功能!因為兩者功能不同!



3. 在使用user space的時候要關掉checksum，原因是在user space的時候rx 與 tx的checksum會發生錯誤，而讓host2收不到封包!

參考資料: <https://mail.openvswitch.org/pipermail/ovs-discuss/2017-November/045764.html>