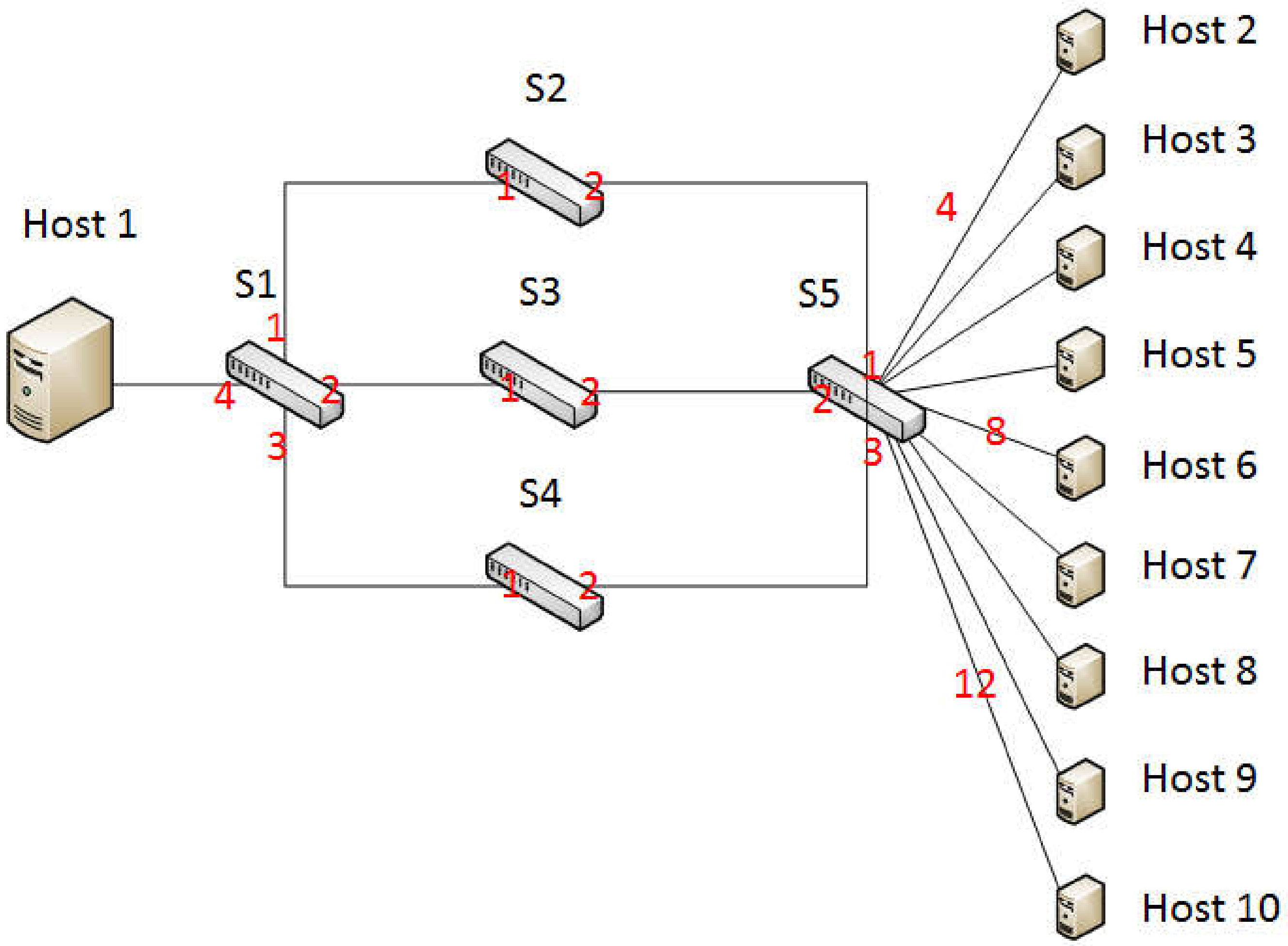
Multipath routing with Group table at mininet

Purpose

在Group table中，有一個類型為`select`，此類型的group會隨機執行底下的其中一個bucket。若我們將所有的output action都放進這個group中，則switch會將封包隨機導向不同的port，藉此達成multipath routing的功用。

Environment

使用下列的圖作為我們的網路環境，在此圖中。S1~S5都是支援OpenFlow 1.3的OpenFlow switch，左邊的Host 1則是一個Sender，會對於右邊的九個Host發送資料



Step

- 使用mininet搭配其script來創造網路拓墣，該script可以在此找到 group.py

```
mn --custom group.py  --topo group
```

- 讓所有的創造的openvswitch都支持openflow 1.3

```
ovs-vsctl set bridge s1 protocols=OpenFlow13
ovs-vsctl set bridge s2 protocols=OpenFlow13
```

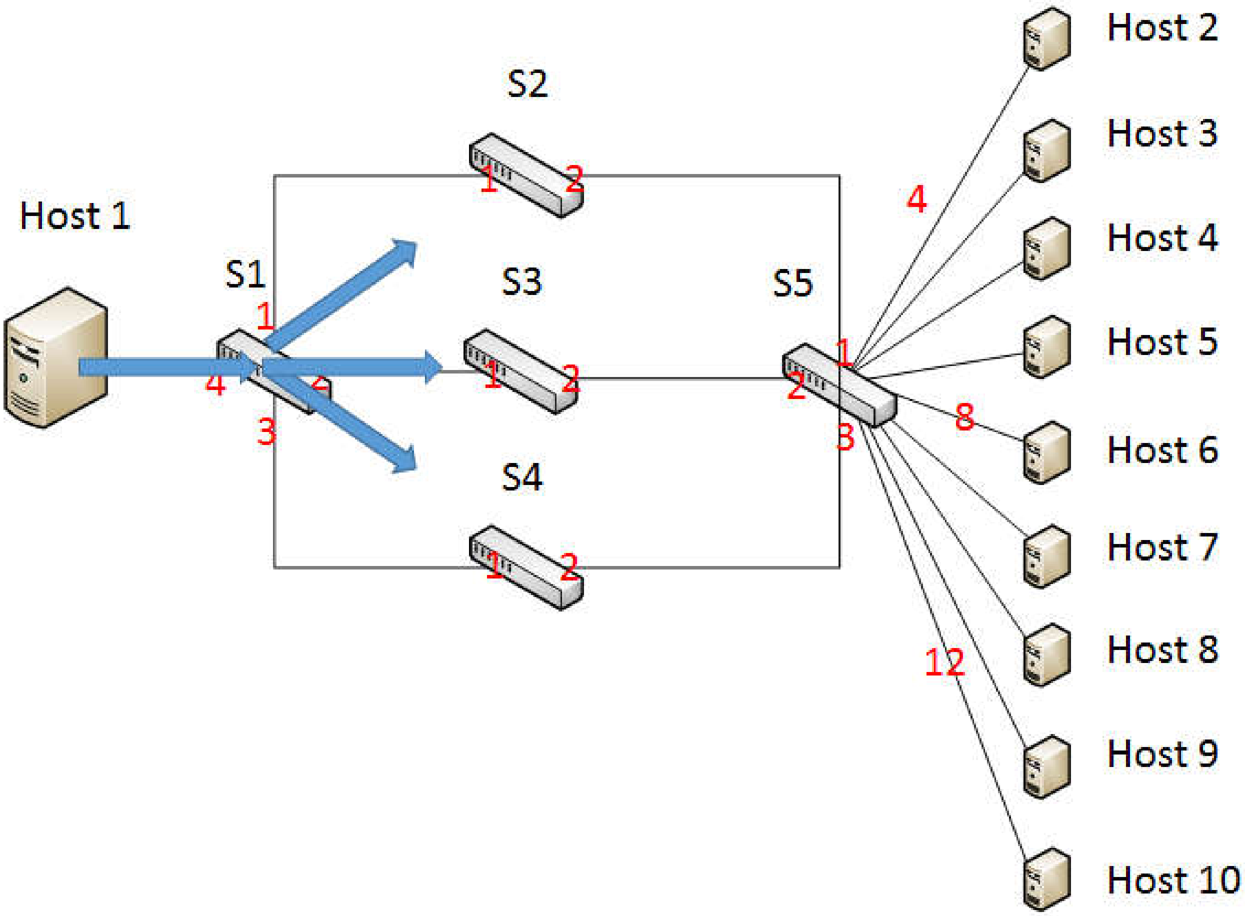
```
ovs-vsctl set bridge s3 protocols=OpenFlow13
ovs-vsctl set bridge s4 protocols=OpenFlow13
ovs-vsctl set bridge s5 protocols=OpenFlow13
```

- 在S1上面加入一個group table，此group table能夠把封包給隨機導向Port 1,2,3。

```
ovs-ofctl -O OpenFlow13 add-group s1 group_id=5566,type=select,bucket=output:
```

- 在S1上面加入一個Flow entry，所有從Host1進來的封包，都去執行剛剛所創立的group table。

```
ovs-ofctl -O OpenFlow13 add-flow s1 in_port=4,actions=group:5566
```



- 由於本實驗沒有採用任何Controller，因此要手動的寫入Flow entry到其餘的Switch。
- 在S1上面加入剩下的Flow entry，使得送回Host1的封包能夠順利抵達Host1

```
ovs-ofctl -O OpenFlow13 add-flow s1 eth_type=0x0800,ip_dst=10.0.0.1,actions=o
ovs-ofctl -O OpenFlow13 add-flow s1 eth_type=0x0806,ip_dst=10.0.0.1,actions=o
```

- 在S2、S3、S4上各加入兩條Flow entry，讓封包能夠通過

```
ovs-ofctl -O OpenFlow13 add-flow s2 in_port=1,actions=output:2
ovs-ofctl -O OpenFlow13 add-flow s2 in_port=2,actions=output:1
ovs-ofctl -O OpenFlow13 add-flow s3 in_port=1,actions=output:2
ovs-ofctl -O OpenFlow13 add-flow s3 in_port=2,actions=output:1
ovs-ofctl -O OpenFlow13 add-flow s4 in_port=1,actions=output:2
ovs-ofctl -O OpenFlow13 add-flow s4 in_port=2,actions=output:1
```

- 在S5上根據destination ip來把封包導向不同的host

```
#IP

ovs-ofctl -O OpenFlow13 add-flow s5 eth_type=0x0800,ip_dst=10.0.0.2,actions=o

ovs-ofctl -O OpenFlow13 add-flow s5 eth_type=0x0800,ip_dst=10.0.0.3,actions=o
ovs-ofctl -O OpenFlow13 add-flow s5 eth_type=0x0800,ip_dst=10.0.0.4,actions=o

ovs-ofctl -O OpenFlow13 add-flow s5 eth_type=0x0800,ip_dst=10.0.0.5,actions=o
ovs-ofctl -O OpenFlow13 add-flow s5 eth_type=0x0800,ip_dst=10.0.0.6,actions=o
ovs-ofctl -O OpenFlow13 add-flow s5 eth_type=0x0800,ip_dst=10.0.0.7,actions=o

ovs-ofctl -O OpenFlow13 add-flow s5 eth_type=0x0800,ip_dst=10.0.0.8,actions=o
```

```
ovs-ofctl -O OpenFlow13 add-flow s5 eth_type=0x0800,ip_dst=10.0.0.9,actions=o
#ARP

ovs-ofctl -O OpenFlow13 add-flow s5 eth_type=0x0806,ip_dst=10.0.0.2,actions=o

ovs-ofctl -O OpenFlow13 add-flow s5 eth_type=0x0806,ip_dst=10.0.0.3,actions=o
ovs-ofctl -O OpenFlow13 add-flow s5 eth_type=0x0806,ip_dst=10.0.0.4,actions=o
ovs-ofctl -O OpenFlow13 add-flow s5 eth_type=0x0806,ip_dst=10.0.0.5,actions=o
ovs-ofctl -O OpenFlow13 add-flow s5 eth_type=0x0806,ip_dst=10.0.0.6,actions=o
ovs-ofctl -O OpenFlow13 add-flow s5 eth_type=0x0806,ip_dst=10.0.0.7,actions=o

ovs-ofctl -O OpenFlow13 add-flow s5 eth_type=0x0806,ip_dst=10.0.0.8,actions=o
ovs-ofctl -O OpenFlow13 add-flow s5 eth_type=0x0806,ip_dst=10.0.0.9,actions=o
```

- 由於本實驗要觀察的是Host1送過來的封包能否走不同路徑，對於送回給Host1的封包就固定於同一條路徑(S5 - S2 - S1)

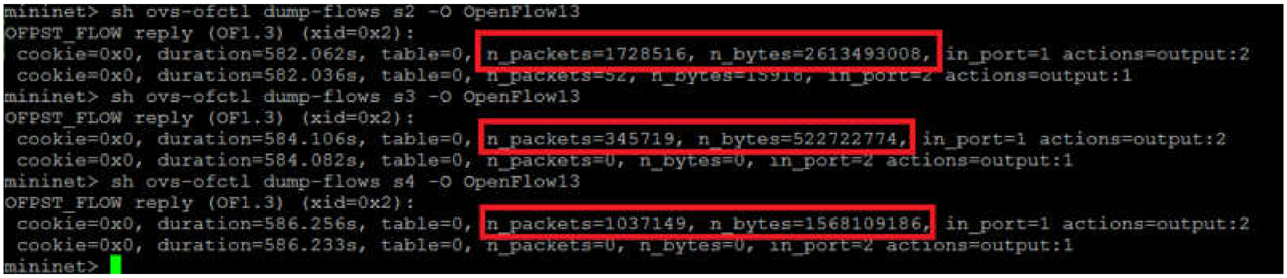
```
ovs-ofctl -O OpenFlow13 add-flow s5 eth_type=0x0800,ip_dst=10.0.0.1,actions=o
ovs-ofctl -O OpenFlow13 add-flow s5 eth_type=0x0806,ip_dst=10.0.0.1,actions=o
```

- 接下來依序執行下列指令來產生網路流量

```
mininet> iperfudp 1G h1 h2
mininet> iperfudp 1G h1 h3
mininet> iperfudp 1G h1 h4
mininet> iperfudp 1G h1 h5
mininet> iperfudp 1G h1 h6
mininet> iperfudp 1G h1 h7
mininet> iperfudp 1G h1 h8
mininet> iperfudp 1G h1 h9
mininet> iperfudp 1G h1 h10
```

- 接下來觀察每個switch的flow table。結果如圖

```
mininet>  sh ovs-ofctl dump-flows s2 -O OpenFlow13
mininet>  sh ovs-ofctl dump-flows s3 -O OpenFlow13
mininet>  sh ovs-ofctl dump-flows s4 -O OpenFlow13
```



```
mininet> sh ovs-ofctl dump-flows s2 -O OpenFlow13
OFPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0x0, duration=582.062s, table=0, n_packets=1728516, n_bytes=2613493008, in_port=1 actions=output:2
 cookie=0x0, duration=582.036s, table=0, n_packets=52, n_bytes=15918, in_port=2 actions=output:1
mininet> sh ovs-ofctl dump-flows s3 -O OpenFlow13
OFPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0x0, duration=584.106s, table=0, n_packets=345719, n_bytes=522722774, in_port=1 actions=output:2
 cookie=0x0, duration=584.082s, table=0, n_packets=0, n_bytes=0, in_port=2 actions=output:1
mininet> sh ovs-ofctl dump-flows s4 -O OpenFlow13
OFPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0x0, duration=586.256s, table=0, n_packets=1037149, n_bytes=1568109186, in_port=1 actions=output:2
 cookie=0x0, duration=586.233s, table=0, n_packets=0, n_bytes=0, in_port=2 actions=output:1
mininet>
```

- 在圖中可以觀察到，S2、S3、S4上面都有流量經過，證實了S1使用了group table會將不同的flow給隨機執行不同的buckets，在此範例中則是會導向不同的port。

Tweet

June 25, 2014

comments powered by Disqus

← OF link capacity ZNC IRC bouncer →