CSEE 4119 Computer Networks

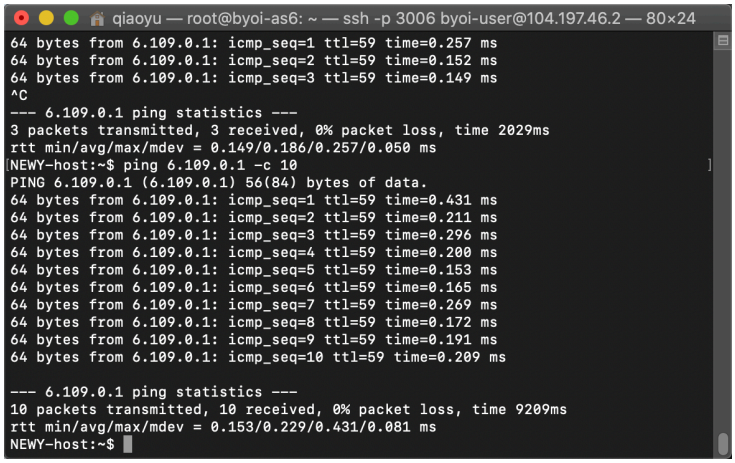
Project 2 Build your own Internet
Stage A

Student: Qiaoyu Gu

UNI: qg2172

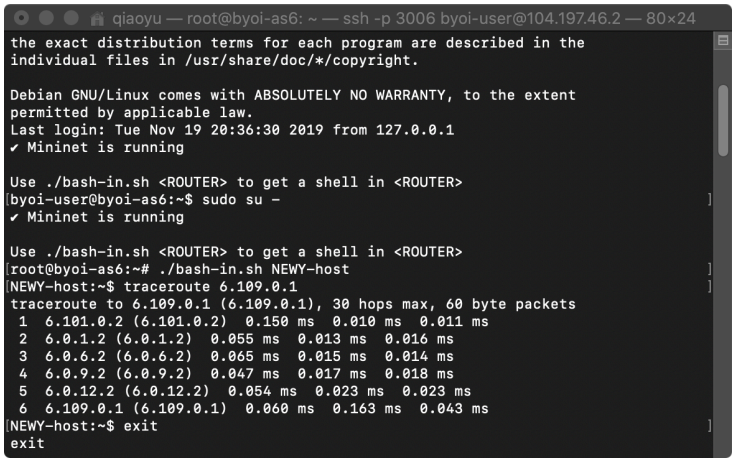
Task 3.2.1: Setup interfaces and set up OSPF

The screenshot of ping results from NEWY-host to SEAT-host by using command **ping 6.109.0.1 -c 10**:



Task 3.2.2: Assign OSPF weights

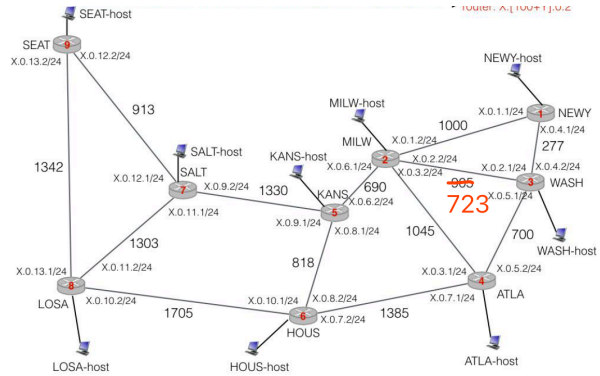
The screenshot of traceroute results from NEWY-host to SEAT-host:



As is shown from the figure above, when connecting to SEAT-host, NEWY-host passes through NEWY, MILW, KANS, SALT, SEAT, SEAT-host sequentially. Such route is the shortest path between two hosts, which corresponds to the mechanism of OSPF. Hence, I have configured OSPF weights properly.

Task 3.2.3: Load balance with ECMP

In this part, I changed OSPF weight between MILW and WASH from 905 to 723. As is shown below:



After making such change, when connecting from NEWY to SEAT, there exists two paths with the same cost of 1000: NEWY- MILW- KANS- SALT- SEAT, NEWY- WASH- MILW- KANS- SALT- SEAT. Due to multiple shortest paths, these routers can split traffic across all of them.

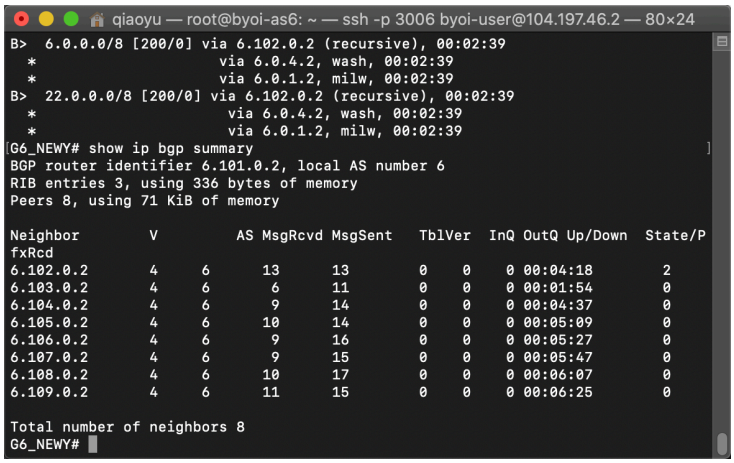
In order to prove ECMP, we can use **show ip route ospf** command. From the result, we can see that when connecting from NEWY to SEAT and SEAT-host, there are equally two paths, via wash or milw separately.

The screenshot of the output of running **show ip route ospf**

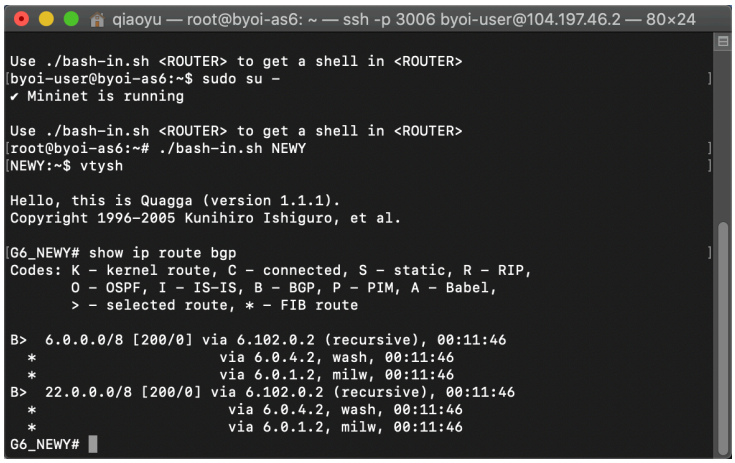
```
G6_NEWY# show ip route ospf
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, P - PIM, A - Babel,
       > - selected route, * - FIB route

O   6.0.1.0/24 [110/1000] is directly connected, milw, 1d00h07m
O>* 6.0.2.0/24 [110/1000] via 6.0.4.2, wash, 21:02:55
O>* 6.0.3.0/24 [110/2022] via 6.0.4.2, wash, 1d00h07m
O   6.0.4.0/24 [110/277] is directly connected, wash, 1d00h07m
O>* 6.0.5.0/24 [110/977] via 6.0.4.2, wash, 1d00h07m
O>* 6.0.6.0/24 [110/1690] via 6.0.4.2, wash, 21:02:55
  *                       via 6.0.1.2, milw, 21:02:55
O>* 6.0.7.0/24 [110/2362] via 6.0.4.2, wash, 1d00h07m
O>* 6.0.8.0/24 [110/2508] via 6.0.4.2, wash, 21:02:55
  *                       via 6.0.1.2, milw, 21:02:55
O>* 6.0.9.0/24 [110/3020] via 6.0.4.2, wash, 21:02:55
  *                       via 6.0.1.2, milw, 21:02:55
O>* 6.0.10.0/24 [110/4067] via 6.0.4.2, wash, 1d00h07m
O>* 6.0.11.0/24 [110/4323] via 6.0.4.2, wash, 21:02:55
  *                       via 6.0.1.2, milw, 21:02:55
O>* 6.0.12.0/24 [110/3933] via 6.0.4.2, wash, 21:02:55
  *                       via 6.0.1.2, milw, 21:02:55
O>* 6.0.13.0/24 [110/5275] via 6.0.4.2, wash, 21:02:55
  *                       via 6.0.1.2, milw, 21:02:55
O   6.101.0.0/24 [110/10] is directly connected, host, 1d00h08m
O>* 6.102.0.0/24 [110/1010] via 6.0.4.2, wash, 21:02:55
  *                         via 6.0.1.2, milw, 21:02:55
O>* 6.103.0.0/24 [110/287] via 6.0.4.2, wash, 1d00h07m
O>* 6.104.0.0/24 [110/987] via 6.0.4.2, wash, 1d00h07m
O>* 6.105.0.0/24 [110/1700] via 6.0.4.2, wash, 21:02:55
  *                         via 6.0.1.2, milw, 21:02:55
O>* 6.106.0.0/24 [110/2372] via 6.0.4.2, wash, 1d00h07m
O>* 6.107.0.0/24 [110/3030] via 6.0.4.2, wash, 21:02:55
  *                         via 6.0.1.2, milw, 21:02:55
O>* 6.108.0.0/24 [110/4077] via 6.0.4.2, wash, 1d00h07m
O>* 6.109.0.0/24 [110/3943] via 6.0.4.2, wash, 21:02:55
  *                         via 6.0.1.2, milw, 21:02:55
G6_NEWY#
```

Task 3.3: Setup iBGP

The screenshot of s**how ip bgp summary** results from NEWY router:



The screenshot of s**how route bgp** results from NEWY router:



When specifying a particular router interface, there needs three steps:

router_name(config-router): neighbor 6.X.0.2 remote-as 6

router_name(config-router): neighbor 6.X.0.2 next-hop-self

router_name(config-router): neighbor 6.X.0.2 update-source 6.Y.0.2

6 is my AS number;

6.X.0.2 is the ip address of host interface of the router we want to connect to under iBGP;

6.Y.0.2 is the ip address of host interface of current router.