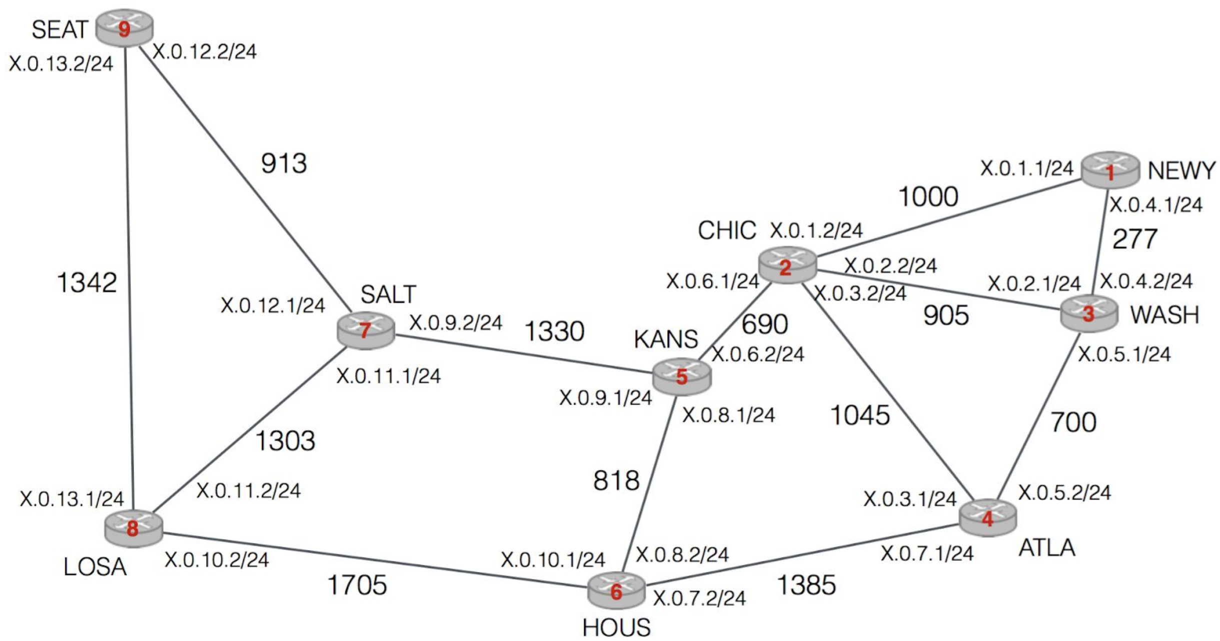
Universidade Federal de Minas Gerais



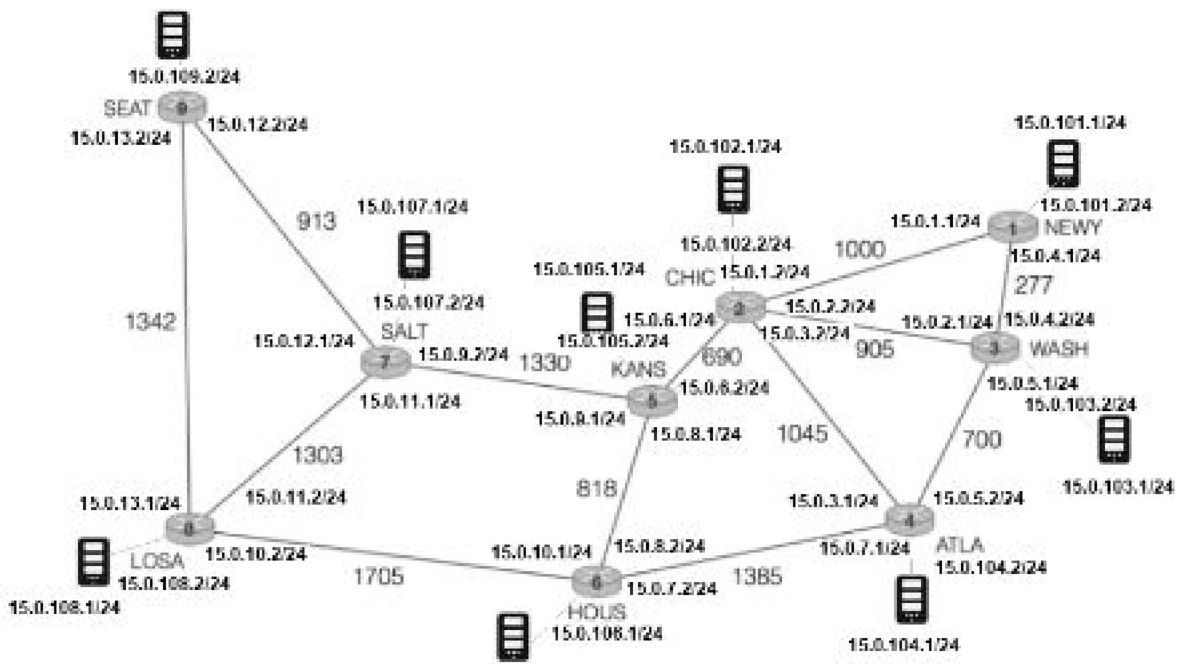
Build your own Internet - Part A

19.04.2018

Otávio Augusto de Oliveira Souza
2018663164

Introduction

This first part request to set the OSPF and iBGp configuration within an AS. The interfaces were properly configured using the CLI in each route and also in each host by setting the correct ip address and default gateway. The network topology is depicted in the figure below:



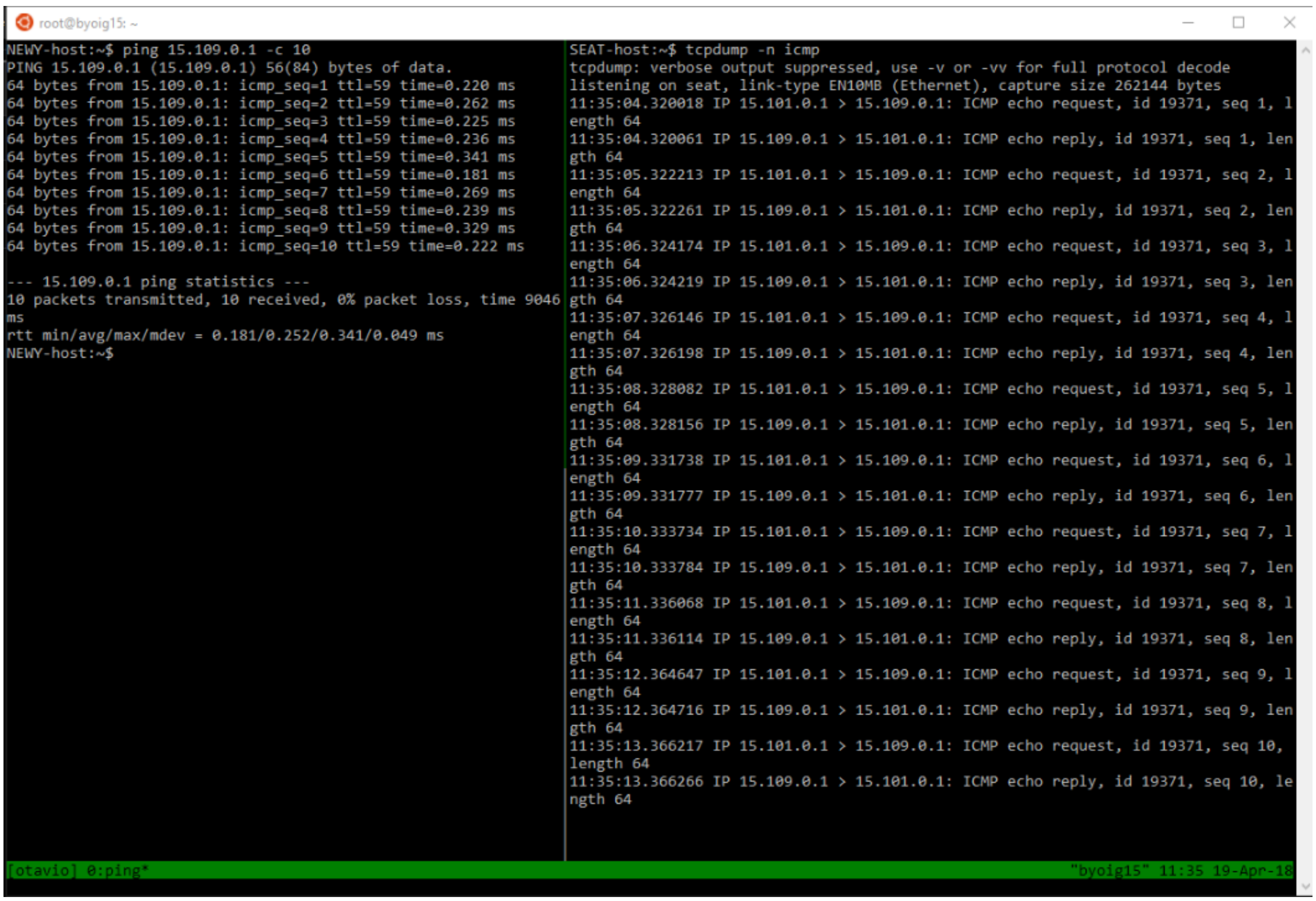
Goals

1. Set the ip address in each host and router interface.
2. Assign properly ospf cost for all enlaces.
3. Enable an ospf session in every router’s prefix.
4. Test ping and traceroute tools between host and router, either by directed or not directed connections.
5. Assess alternative cost to path.
6. Assign iBGP session.

Question 1

In this part evaluates a ping request between each pair of host and router node. The results show that the network are properly connect and all nodes are achievable. As required on the assignment, below is the screenshot of 10 ping results from NEWY-host to SEAT-host.

Screenshot - Ping results



```
NEWY-host:~$ ping 15.109.0.1 -c 10
PING 15.109.0.1 (15.109.0.1) 56(84) bytes of data.
64 bytes from 15.109.0.1: icmp_seq=1 ttl=59 time=0.220 ms
64 bytes from 15.109.0.1: icmp_seq=2 ttl=59 time=0.262 ms
64 bytes from 15.109.0.1: icmp_seq=3 ttl=59 time=0.225 ms
64 bytes from 15.109.0.1: icmp_seq=4 ttl=59 time=0.236 ms
64 bytes from 15.109.0.1: icmp_seq=5 ttl=59 time=0.341 ms
64 bytes from 15.109.0.1: icmp_seq=6 ttl=59 time=0.181 ms
64 bytes from 15.109.0.1: icmp_seq=7 ttl=59 time=0.269 ms
64 bytes from 15.109.0.1: icmp_seq=8 ttl=59 time=0.239 ms
64 bytes from 15.109.0.1: icmp_seq=9 ttl=59 time=0.329 ms
64 bytes from 15.109.0.1: icmp_seq=10 ttl=59 time=0.222 ms

--- 15.109.0.1 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9046
ms
rtt min/avg/max/mdev = 0.181/0.252/0.341/0.049 ms
NEWY-host:~$
```

```
SEAT-host:~$ tcpdump -n icmp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on seat, link-type EN10MB (Ethernet), capture size 262144 bytes
11:35:04.320018 IP 15.101.0.1 > 15.109.0.1: ICMP echo request, id 19371, seq 1, length 64
11:35:04.320061 IP 15.109.0.1 > 15.101.0.1: ICMP echo reply, id 19371, seq 1, length 64
11:35:05.322213 IP 15.101.0.1 > 15.109.0.1: ICMP echo request, id 19371, seq 2, length 64
11:35:05.322261 IP 15.109.0.1 > 15.101.0.1: ICMP echo reply, id 19371, seq 2, length 64
11:35:06.324174 IP 15.101.0.1 > 15.109.0.1: ICMP echo request, id 19371, seq 3, length 64
11:35:06.324219 IP 15.109.0.1 > 15.101.0.1: ICMP echo reply, id 19371, seq 3, length 64
11:35:07.326146 IP 15.101.0.1 > 15.109.0.1: ICMP echo request, id 19371, seq 4, length 64
11:35:07.326198 IP 15.109.0.1 > 15.101.0.1: ICMP echo reply, id 19371, seq 4, length 64
11:35:08.328082 IP 15.101.0.1 > 15.109.0.1: ICMP echo request, id 19371, seq 5, length 64
11:35:08.328156 IP 15.109.0.1 > 15.101.0.1: ICMP echo reply, id 19371, seq 5, length 64
11:35:09.331738 IP 15.101.0.1 > 15.109.0.1: ICMP echo request, id 19371, seq 6, length 64
11:35:09.331777 IP 15.109.0.1 > 15.101.0.1: ICMP echo reply, id 19371, seq 6, length 64
11:35:10.333734 IP 15.101.0.1 > 15.109.0.1: ICMP echo request, id 19371, seq 7, length 64
11:35:10.333784 IP 15.109.0.1 > 15.101.0.1: ICMP echo reply, id 19371, seq 7, length 64
11:35:11.336068 IP 15.101.0.1 > 15.109.0.1: ICMP echo request, id 19371, seq 8, length 64
11:35:11.336114 IP 15.109.0.1 > 15.101.0.1: ICMP echo reply, id 19371, seq 8, length 64
11:35:12.364647 IP 15.101.0.1 > 15.109.0.1: ICMP echo request, id 19371, seq 9, length 64
11:35:12.364716 IP 15.109.0.1 > 15.101.0.1: ICMP echo reply, id 19371, seq 9, length 64
11:35:13.366217 IP 15.101.0.1 > 15.109.0.1: ICMP echo request, id 19371, seq 10, length 64
11:35:13.366266 IP 15.109.0.1 > 15.101.0.1: ICMP echo reply, id 19371, seq 10, length 64
```


Question 2

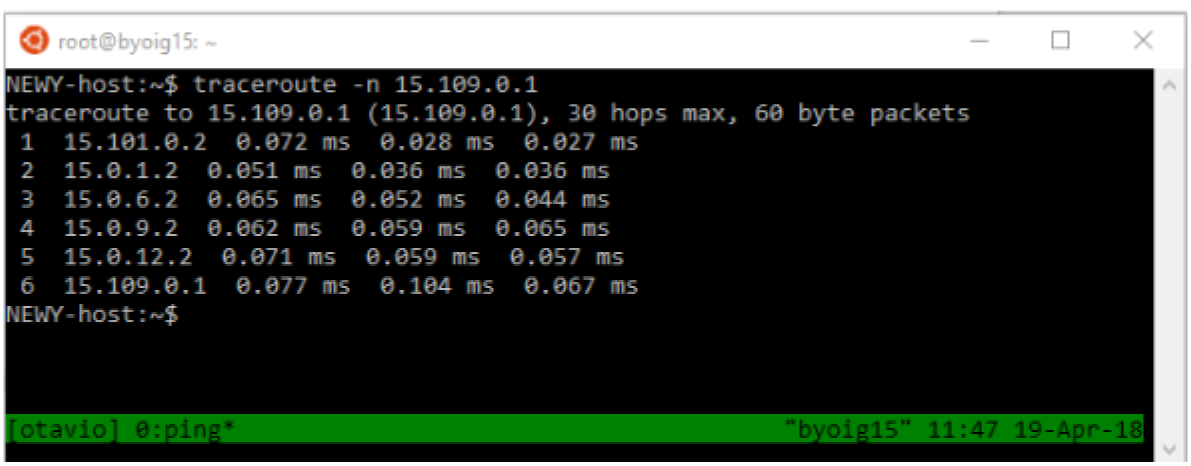
This part assesses the connections and evaluate the configuration of OSPF protocol. A bunch of traceroute request was made between each pair of router and host in order to see if chosen paths was correct. To proof the correctness of the results, a Dijkstra's algorithm was run over the topology graph and the result can be see in the table below.

The Dijkstra using NEWY as the source:

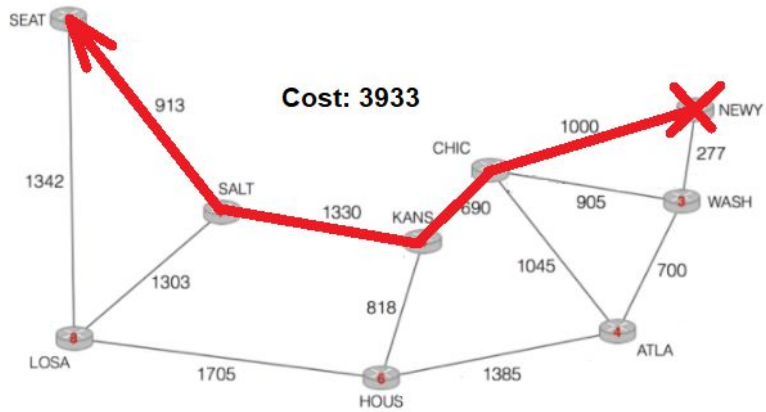
Nodes x min-path priority-queue	1	2	3	4	5	6	7	8	9
1	0	1000	277	INF	INF	INF	INF	INF	INF
3	0	1000	277	977	INF	INF	INF	INF	INF
4	0	1000	277	977	INF	2362	INF	INF	INF
2	0	1000	277	977	1690	2362	INF	INF	INF
5	0	1000	277	977	1690	2362	3020	INF	INF
6	0	1000	277	977	1690	2362	3020	4067	INF
7	0	1000	277	977	1690	2362	3020	4067	3933
9	0	1000	277	977	1690	2362	3020	4067	**3933**

1.Dijkstra's table between node 1(NEWY) and node 9(SEAT)

The screenshot of a traceroute from the host in NEWY to the host in SEAT :



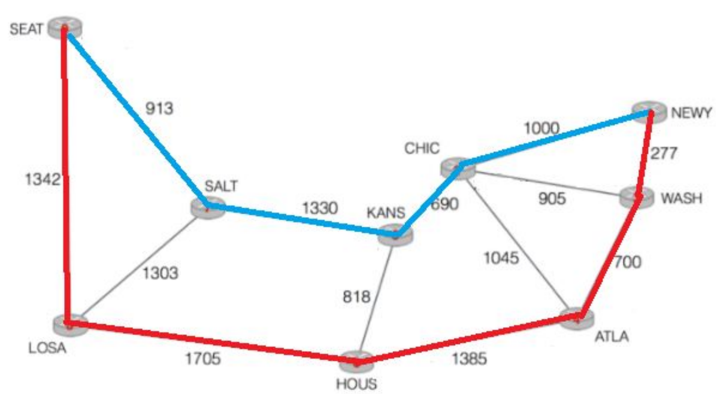
The path chosen according to traceroute:



Question 3

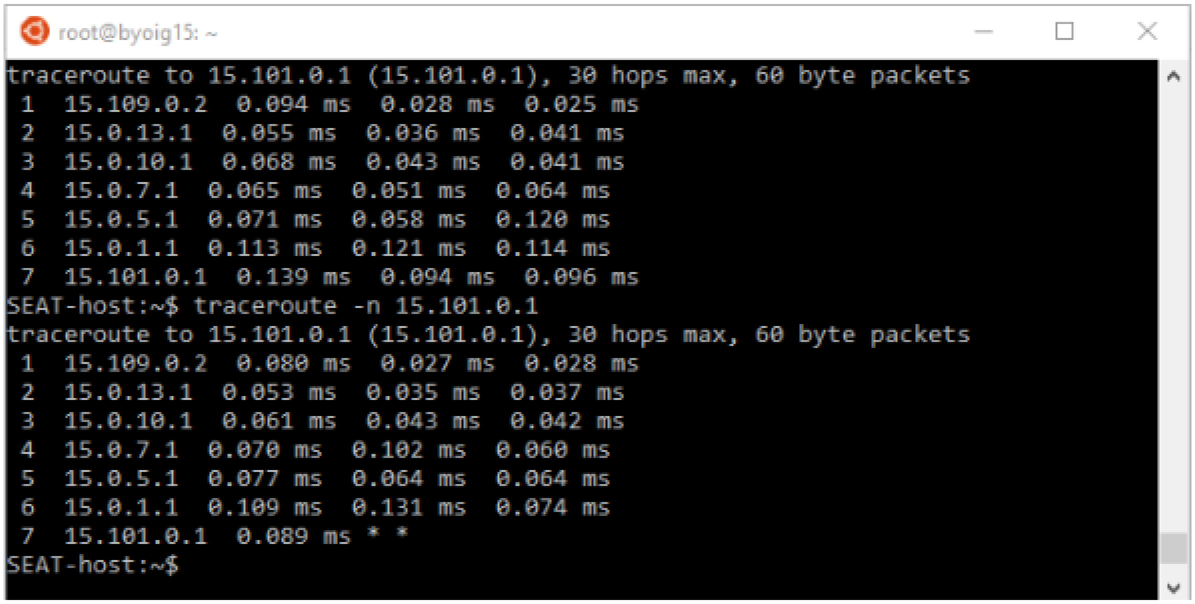
The question 3 assumes that a huge amount of traffic was continuously sent from SEAT to NEWY, producing congestion in some of the links between these two routers. This problem can be solved using a disjoint path between these two routers. A solution that split the traffic in some point could work well, but I assume that the proposal idea is to find a disjoint path that avoid the overload of the links.

The only disjoint path available is show by the pictures below:



So, one possible solution would be increasing the cost of the link KANS-CHIC to make the two path equal in terms of cost. The cost of the link changed of 690 to 2166, as consequence these two path became with the same cost of 5409. Even though the cost increases, the overload traffic won't happen due to the disjoint paths.

The Experiments:



As shown above, the path follow by the traceroute forward path is different from the backward path due to the interface that that message was sent back (15.0.1.1) do not correspond with the interface that the message arrived (15.0.4.1). This prove that the path is disjoint and could not interfere in its owns traffic.

Question 4

In this final question a BGP session (iBGP) between all pairs of routers (full-mesh) was configured. And the screenshot of NEWY router are shown below. Particularly, the ip address used in any session corresponds with the ip address between the interface with each host. This is done in order to avoid unnecessary losses in sessions due to interface's failures.

The commands used:

router_name#conf t
rotuer_name#router bgp 15
router_name(config-router)#neighbor <neighbor-ip-interface-with-host> remote-as 15
…

root@byoig15: ~

```
G15_NEWY# show ip bgp summary
BGP router identifier 15.101.0.2, local AS number 15
RIB entries 0, using 0 bytes of memory
Peers 8, using 71 KiB of memory

Neighbor        V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/P
fxRcd
15.102.0.2      4    15       0      75        0    0    0 never    Active
15.103.0.2      4    15       0      76        0    0    0 never    Active
15.104.0.2      4    15       0      76        0    0    0 never    Active
15.105.0.2      4    15       0      77        0    0    0 never    Active
15.106.0.2      4    15       0      77        0    0    0 never    Active
15.107.0.2      4    15       0      77        0    0    0 never    Active
15.108.0.2      4    15       0      77        0    0    0 never    Active
15.109.0.2      4    15       0      78        0    0    0 never    Active

Total number of neighbors 8
G15_NEWY#
```