



دانشکده مهندسی کامپیوتر

آزمایشگاه شبکه‌های کامپیوتری

گزارش کار آزمایش ۷ - بخش دوم

گروه ۴

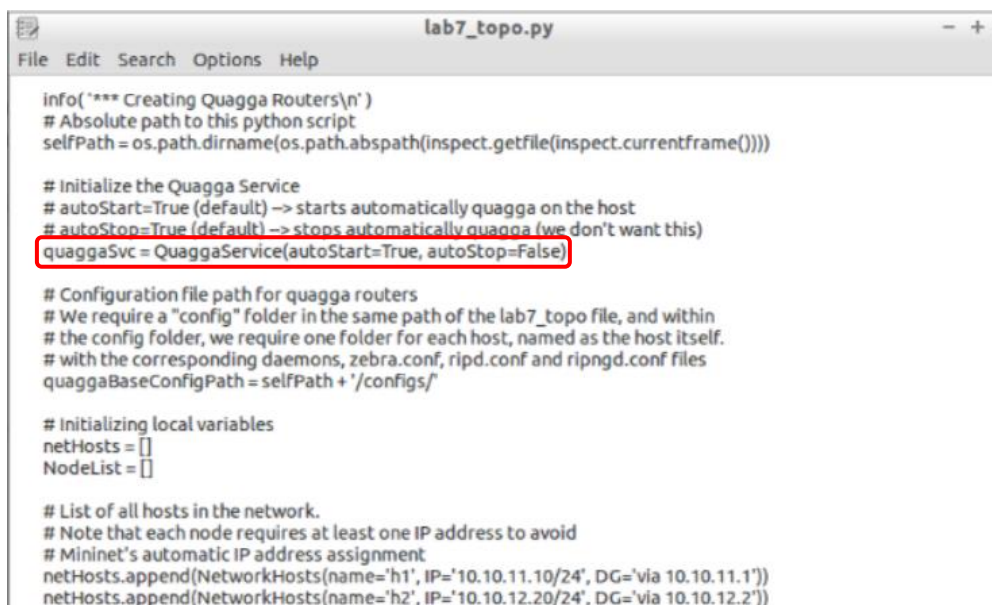
علی صداقی ۹۷۵۲۱۳۷۸

دانیال بازمانده ۹۷۵۲۱۱۳۵

۱ بخش ج-۲

فایل مربوط به ساخت شبکه‌ی lab7_network.py را مجدداً اجرا می‌کنیم تا شبکه ساخته شود. سپس سرویس quagga را در تمامی روترها ریست می‌کنیم.

برای انجام این کار، در فایل lab7_topo.py کاری می‌کنیم که سرویس quagga به طور خودکار شروع شود. این کار را با تغییر خط مشخص شده در شکل زیر انجام می‌دهیم و مقدار autoStart را برابر True قرار می‌دهیم.



```
lab7_topo.py
File Edit Search Options Help

info(*** Creating Quagga Routers\n')
# Absolute path to this python script
selfPath = os.path.dirname(os.path.abspath(inspect.getfile(inspect.currentframe())))

# Initialize the Quagga Service
# autoStart=True (default) -> starts automatically quagga on the host
# autoStop=True (default) -> stops automatically quagga (we don't want this)
quaggaSvc = QuaggaService(autoStart=True, autoStop=False)

# Configuration file path for quagga routers
# We require a "config" folder in the same path of the lab7_topo file, and within
# the config folder, we require one folder for each host, named as the host itself.
# with the corresponding daemons, zebra.conf, ripd.conf and ripngd.conf files
quaggaBaseConfigPath = selfPath + '/configs/'

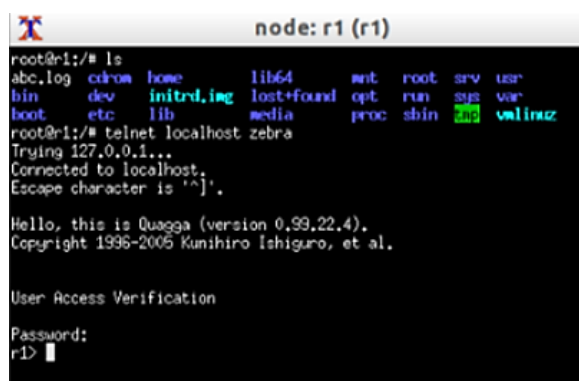
# Initializing local variables
netHosts = []
NodeList = []

# List of all hosts in the network.
# Note that each node requires at least one IP address to avoid
# Mininet's automatic IP address assignment
netHosts.append(NetworkHosts(name='h1', IP='10.10.11.10/24', DG='via 10.10.11.1'))
netHosts.append(NetworkHosts(name='h2', IP='10.10.12.20/24', DG='via 10.10.12.2'))
```

برای مانیتور کردن فعالیت یک پردازنده در Quagga از طریق telnet تلاش می‌کنیم به پروسس مربوطه متصل شویم و وارد مانیتورینگ شویم. این کار را با استفاده از دستور زیر انجام می‌دهیم:

telnet localhost zebra

این دستور را در ترمینال مربوط به r1 اجرا می‌کنیم تا به پروسس zebra متصل شویم.



```
node: r1 (r1)
root@r1:~# ls
abc.log  cdrom  home  lib64  ant  root  srv  usr
bin      dev    initrd.img  lost+found  opt  run  sfs  var
boot     etc    lib      media  proc  shin  top  vulnuz

root@r1:~# telnet localhost zebra
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Hello, this is Quagga (version 0.99.22.4).
Copyright 1996-2005 Kunihiko Ishiguro, et al.

User Access Verification

Password:
r1>
```

همانطور که مشاهده می‌شود، این اتصال موفقیت‌آمیز بوده است.

۲ سوال ۲

در مرحله بعد، با استفاده از دستور `show ip route` به محتوای جداول مسیریابی IPv4 دسترسی پیدا می‌کنیم.

```
r1> show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

C>* 10.10.11.0/24 is directly connected, r1-eth3
C>* 10.10.12.0/24 is directly connected, r1-eth1
C>* 10.10.14.0/24 is directly connected, r1-eth2
C>* 127.0.0.0/8 is directly connected, lo
r1>
```

همانطور که در تصویر مشاهده می‌شود، همانند آنچه مطابق تصویر توپولوژی انتظار داشتیم، روتر مربوطه سه سابنت `eth1-eth2-eth3` را داراست.

۳ سوال ۳

در مرحله بعد، با استفاده از دستور `show interface` روی ترمینال r3 وضعیت اینترفیس‌های موجود را بررسی می‌کنیم.

```
node: r3 (r3)
User Access Verification
Password:
r3> show interface
Interface lo is up, line protocol detection is disabled
  index 1 metric 1 mtu 65536
  flags: <UP,LOOPBACK,RUNNING>
  inet 127.0.0.1/8
  inet6 ::1/128
Interface r3-eth1 is up, line protocol detection is disabled
  index 90 metric 1 mtu 1500
  flags: <UP,BROADCAST,RUNNING,MULTICAST>
  HWaddr: f6:28:bd:e3:17:c4
  inet 10.10.23.3/24 broadcast 10.10.23.255
  inet6 fe80::f428:bdff:fee3:17c4/64
Interface r3-eth2 is up, line protocol detection is disabled
  index 104 metric 1 mtu 1500
  flags: <UP,BROADCAST,RUNNING,MULTICAST>
  HWaddr: 36:b7:55:51:6a:0f
  inet 10.10.35.3/24 broadcast 10.10.35.255
  inet6 fe80::34b7:55ff:fe51:6a0f/64
r3>
```

همانطور که در تصویر مشاهده می‌شود، همانند آنچه مطابق تصویر توپولوژی انتظار داشتیم، روتر مربوطه دو اینترفیس `eth1-eth2` را داراست.

۴ سوال ۴

در مرحله بعد، ابتدا با استفاده از دستور enable مد پیکربندی را فعال می‌کنیم و با استفاده از دستور configure terminal وارد مد پیکربندی می‌شویم.

سپس با تایپ دستور interface r4-eth2 وارد مد پیکربندی اینترفیس eth2 می‌شویم. همچنین می‌توانیم با استفاده از دستور ip address 10.10.45.4/24 به اینترفیس مربوطه آدرس IP ورژن ۴ اختصاص دهیم.

در انتها با دو بار تایپ exit از مد پیکربندی اینترفیس و سپس از مد پیکربندی خارج می‌شویم.

در نهایت با استفاده از دستور show interface از دستورات تغییرات داده‌شده مطمئن می‌شویم که همانطور که مشاهده می‌شود، تغییرات به درستی صورت گرفته است و آدرس IP مربوط به اینترفیس eth2 تغییر کرده است.

```
node: r4 (r4)
r4# configure terminal
r4(config)# interface r4-eth2
r4(config-if)# ip address 10.10.45.4/24
r4(config-if)# exit
r4(config)# exit
r4# show interface
Interface lo is up, line protocol detection is disabled
  index 1 metric 1 mtu 65536
  flags: <UP,LOOPBACK,RUNNING>
  inet 127.0.0.1/8
  inet6 ::1/128
Interface r4-eth1 is up, line protocol detection is disabled
  index 94 metric 1 mtu 1500
  flags: <UP,BROADCAST,RUNNING,MULTICAST>
  Hwaddr: 6a:d6:61:3b:30:a8
  inet 10.10.14.4/24 broadcast 10.10.14.255
  inet6 fe80::68d6:61ff:fe3b:30a8/64
Interface r4-eth2 is up, line protocol detection is disabled
  index 100 metric 1 mtu 1500
  flags: <UP,BROADCAST,RUNNING,MULTICAST>
  Hwaddr: f6:f4:ca:d5:b5:38
  inet 10.10.45.4/24 broadcast 10.10.45.255
  inet6 fe80::f4f4:caff:fed5:b538/64
r4#
```