



دانشکده مهندسی کامپیوتر
آزمایشگاه شبکه‌های کامپیوتری

گزارش کار آزمایش ۸

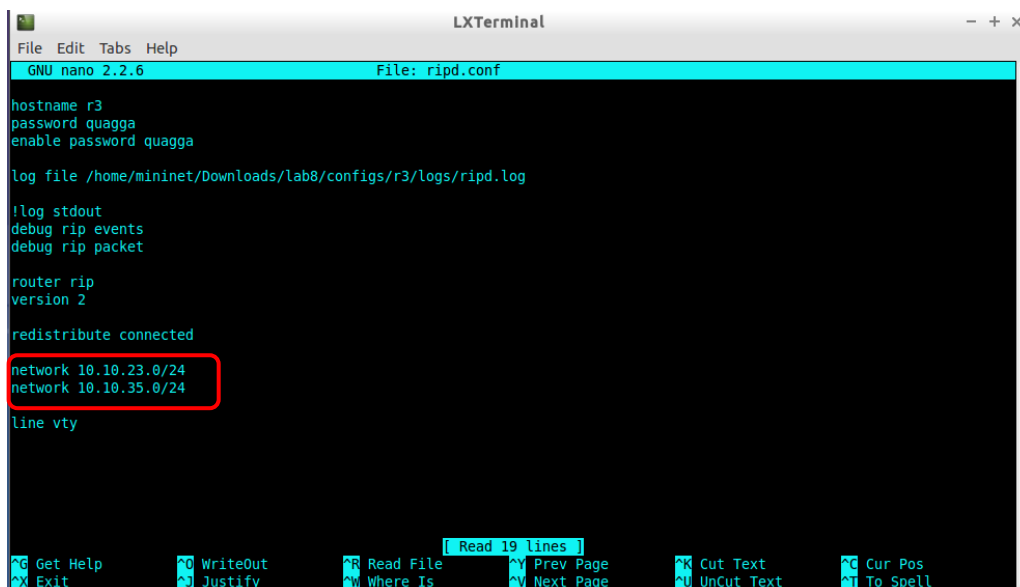
گروه ۴

علی صدیقی ۹۷۵۲۱۳۷۸

دانیال بازمانده ۹۷۵۲۱۱۳۵

۱ بخش الف

ابتدا اقدام به کپی کردن فایل‌های ripd.conf از روتر r1 برای هر یک از روترهای r3 و r5 می‌کنیم و در هر کدام، پیکربندی لازم را برای ripd ایجاد می‌کنیم و در واقع هر یک از network ها را مطابق شکل توپولوژی داده‌شده تنظیم می‌کنیم.



```
File Edit Tabs Help
GNU nano 2.2.6 File: ripd.conf

hostname r3
password quagga
enable password quagga

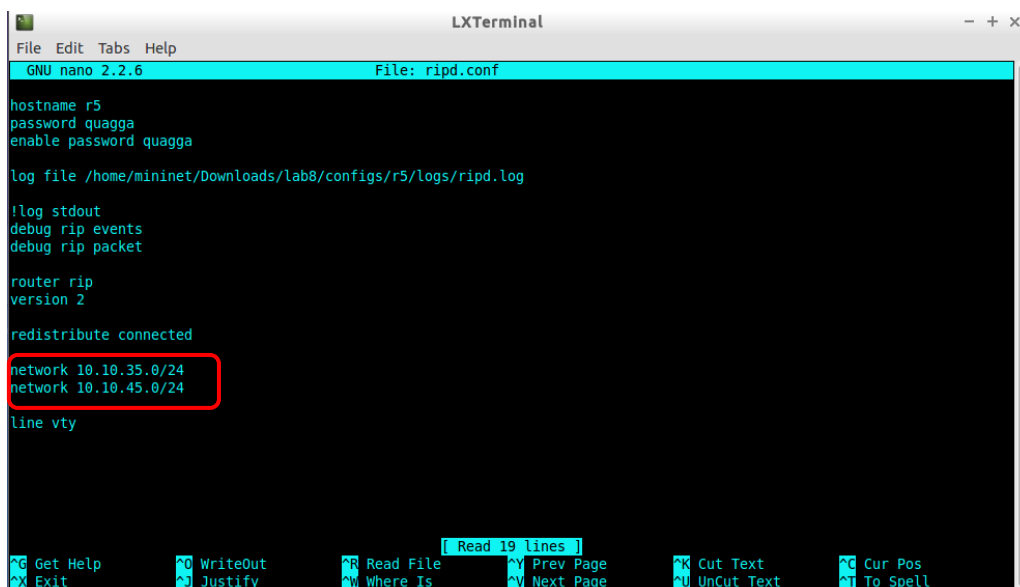
log file /home/mininet/Downloads/lab8/configs/r3/logs/ripd.log

!log stdout
debug rip events
debug rip packet

router rip
version 2

redistribute connected
network 10.10.23.0/24
network 10.10.35.0/24
line vty

^G Get Help      ^O WriteOut      ^R Read File      ^Y Prev Page      ^K Cut Text      ^C Cur Pos
^X Exit          ^J Justify       ^W Where Is      ^V Next Page      ^U UnCut Text    ^T To Spell
```



```
File Edit Tabs Help
GNU nano 2.2.6 File: ripd.conf

hostname r5
password quagga
enable password quagga

log file /home/mininet/Downloads/lab8/configs/r5/logs/ripd.log

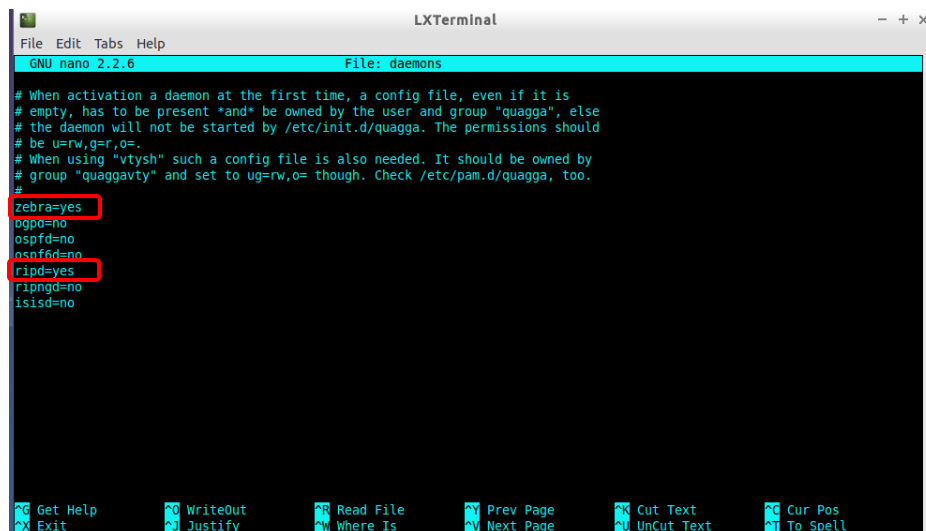
!log stdout
debug rip events
debug rip packet

router rip
version 2

redistribute connected
network 10.10.35.0/24
network 10.10.45.0/24
line vty

^G Get Help      ^O WriteOut      ^R Read File      ^Y Prev Page      ^K Cut Text      ^C Cur Pos
^X Exit          ^J Justify       ^W Where Is      ^V Next Page      ^U UnCut Text    ^T To Spell
```

سپس با بررسی محتوای فایل daemons بررسی می‌کنیم که پروسس‌های zebra و ripd حتما فعال باشند.



```

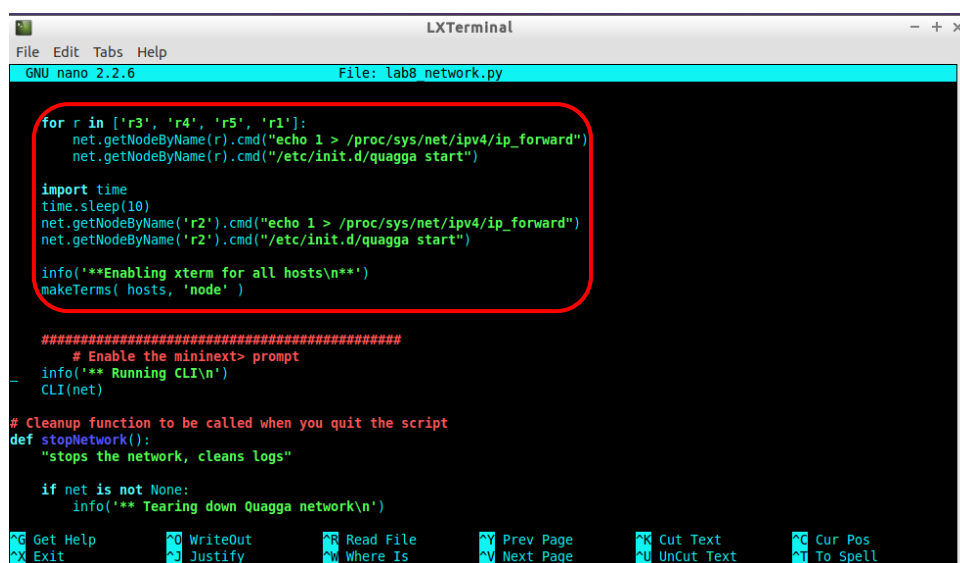
GNU nano 2.2.6 File: daemons

# When activation a daemon at the first time, a config file, even if it is
# empty, has to be present "and" be owned by the user and group "quagga", else
# the daemon will not be started by /etc/init.d/quagga. The permissions should
# be u=rw,g=r,o=.
# When using "vtysh" such a config file is also needed. It should be owned by
# group "quagga" and set to ug=rw,o= though. Check /etc/pam.d/quagga, too.
#
zebra=yes
bgpd=no
ospfd=no
nsnfd=no
ripd=yes
ripngd=no
isisd=no

Get Help WriteOut Read File Prev Page Cut Text Cur Pos
Exit Justify Where Is Next Page UnCut Text To Spell

```

در مرحله بعد، با دستکاری فایل lab8_network.py اطمینان حاصل می‌کنیم که روترها به ترتیب r3,r4,r5,r1 راه‌اندازی شوند و پس از ۱۰ ثانیه سرویس quagga روی r2 استارت شود.



```

GNU nano 2.2.6 File: lab8_network.py

for r in ['r3', 'r4', 'r5', 'r1']:
    net.getNodeByName(r).cmd("echo 1 > /proc/sys/net/ipv4/ip_forward")
    net.getNodeByName(r).cmd("/etc/init.d/quagga start")

import time
time.sleep(10)
net.getNodeByName('r2').cmd("echo 1 > /proc/sys/net/ipv4/ip_forward")
net.getNodeByName('r2').cmd("/etc/init.d/quagga start")

info('**Enabling xterm for all hosts\n**')
makeTerms( hosts, 'node' )

#####
# Enable the mininet> prompt
info('** Running CLI\n')
CLI(net)

# Cleanup function to be called when you quit the script
def stopNetwork():
    "stops the network, cleans logs"

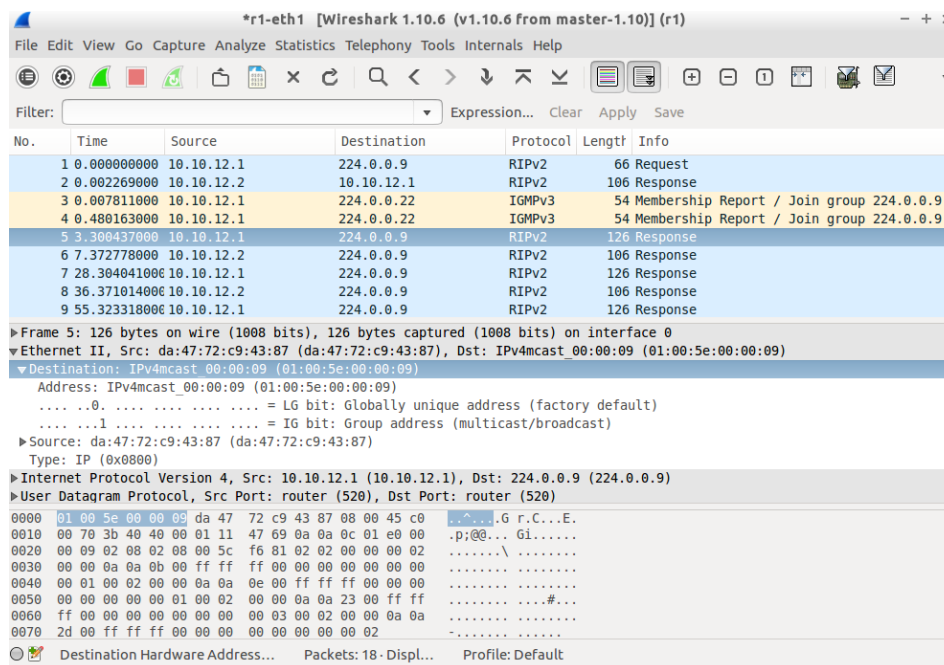
    if net is not None:
        info('** Tearing down Quagga network\n')

Get Help WriteOut Read File Prev Page Cut Text Cur Pos
Exit Justify Where Is Next Page UnCut Text To Spell

```

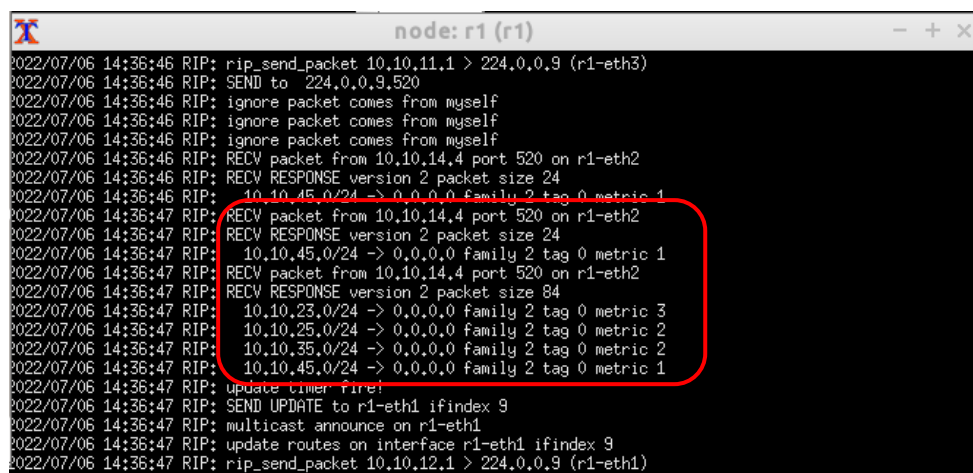
۲ سوال ۱

با راه اندازی Wireshark روی r1-eth1 متوجه می شویم که اطلاعات با الگوی چندبخشی (multicast) بین روترها دست به دست می شوند.



۳ سوال ۲

خط متناظر با Wireshark در فایل لاگ ها به صورت زیر است.



۴ سوال ۳

در لحظه‌ی 14:36:47 روتر r1 موفق به دریافت نخستین پیام از r4 گردیده است. شبکه‌های RIPv2 در این پیام تبلیغ شده‌اند.

```
2022/07/06 14:36:47 RIP: 10.10.45.0/24 -> 0.0.0.0 family 2 tag 0 metric 1
2022/07/06 14:36:47 RIP: RECV packet from 10.10.14.4 port 520 on r1-eth2
2022/07/06 14:36:47 RIP: RECV RESPONSE version 2 packet size 84
2022/07/06 14:36:47 RIP: 10.10.23.0/24 -> 0.0.0.0 family 2 tag 0 metric 3
2022/07/06 14:36:47 RIP: 10.10.25.0/24 -> 0.0.0.0 family 2 tag 0 metric 2
2022/07/06 14:36:47 RIP: 10.10.35.0/24 -> 0.0.0.0 family 2 tag 0 metric 2
2022/07/06 14:36:47 RIP: 10.10.45.0/24 -> 0.0.0.0 family 2 tag 0 metric 1
2022/07/06 14:36:47 RIP: update timer fire!
2022/07/06 14:36:47 RIP: SEND UPDATE to r1-eth1 ifindex 9
2022/07/06 14:36:47 RIP: multicast announce on r1-eth1
```

۵ سوال ۴

در لحظه‌ی 14:36:56 روتر r1 موفق به دریافت نخستین پیام از r2 گردیده است. شبکه‌های RIPv2 در این پیام تبلیغ شده‌اند.

```
2022/07/06 14:36:47 RIP: update routes on interface r1-eth1 ifindex 9
2022/07/06 14:36:47 RIP: rip_send_packet 10.10.12.1 > 224.0.0.9 (r1-eth1)
2022/07/06 14:36:47 RIP: SEND to 224.0.0.9,520
2022/07/06 14:36:47 RIP: SEND RESPONSE version 2 packet size 124
2022/07/06 14:36:47 RIP: 10.10.11.0/24 -> 0.0.0.0 family 2 tag 0 metric 1
2022/07/06 14:36:47 RIP: 10.10.14.0/24 -> 0.0.0.0 family 2 tag 0 metric 1
2022/07/06 14:36:47 RIP: 10.10.23.0/24 -> 0.0.0.0 family 2 tag 0 metric 4
2022/07/06 14:36:47 RIP: 10.10.25.0/24 -> 0.0.0.0 family 2 tag 0 metric 3
2022/07/06 14:36:47 RIP: 10.10.35.0/24 -> 0.0.0.0 family 2 tag 0 metric 3
2022/07/06 14:36:47 RIP: 10.10.45.0/24 -> 0.0.0.0 family 2 tag 0 metric 2
2022/07/06 14:36:47 RIP: SEND UPDATE to r1-eth2 ifindex 17
```

۶ سوال ۵

همانطور که مشاهده می‌شود، به تعداد ۴ پیام ZEBRA_IPV4_ROUTE_ADD در 14:36:45 داریم. با توجه به اینکه r1 مسیره‌های بهتری برای شبکه‌ها ندارد، همگی آنها پذیرفته شده‌اند.

```
2022/07/06 14:36:45 ZEBRA: zebra message received [ZEBRA_REDISTRIBUTE_ADD] 1
2022/07/06 14:36:46 ZEBRA: zebra message received [ZEBRA_IPV4_ROUTE_ADD] 19
2022/07/06 14:36:47 ZEBRA: zebra message received [ZEBRA_IPV4_ROUTE_ADD] 19
2022/07/06 14:36:47 ZEBRA: zebra message received [ZEBRA_IPV4_ROUTE_ADD] 19
2022/07/06 14:36:47 ZEBRA: zebra message received [ZEBRA_IPV4_ROUTE_ADD] 19
2022/07/06 14:36:56 ZEBRA: zebra message received [ZEBRA_IPV4_ROUTE_DELETE] 19
2022/07/06 14:36:56 ZEBRA: zebra message received [ZEBRA_IPV4_ROUTE_ADD] 19
2022/07/06 14:36:56 ZEBRA: zebra message received [ZEBRA_IPV4_ROUTE_DELETE] 19
2022/07/06 14:36:56 ZEBRA: zebra message received [ZEBRA_IPV4_ROUTE_ADD] 19
2022/07/06 14:42:53 ZEBRA: zebra message received [ZEBRA_IPV4_ROUTE_DELETE] 19
2022/07/06 14:42:53 ZEBRA: zebra message received [ZEBRA_IPV4_ROUTE_DELETE] 19
2022/07/06 14:42:53 ZEBRA: zebra message received [ZEBRA_IPV4_ROUTE_DELETE] 19
2022/07/06 14:42:53 ZEBRA: zebra message received [ZEBRA_IPV4_ROUTE_DELETE] 19
```

۷ سوال ۶

همانطور که مشاهده می‌شود، به تعداد ۲ پیام ZEBRA_IPV4_ROUTE_ADD در 14:36:45 داریم. با توجه به اینکه r2 hop های کمتری به 10.10.23.0/24 دارد، پیام قبلی به r4 حذف می‌شود و پیام جدید به r2 اضافه می‌شود.

```
2022/07/06 14:36:45 ZEBRA: zebra message received [ZEBRA_REDISTRIBUTE_ADD] 1
2022/07/06 14:36:46 ZEBRA: zebra message received [ZEBRA_IPV4_ROUTE_ADD] 19
2022/07/06 14:36:47 ZEBRA: zebra message received [ZEBRA_IPV4_ROUTE_ADD] 19
2022/07/06 14:36:47 ZEBRA: zebra message received [ZEBRA_IPV4_ROUTE_ADD] 19
2022/07/06 14:36:47 ZEBRA: zebra message received [ZEBRA_IPV4_ROUTE_ADD] 19
2022/07/06 14:36:56 ZEBRA: zebra message received [ZEBRA_IPV4_ROUTE_DELETE] 19
2022/07/06 14:36:56 ZEBRA: zebra message received [ZEBRA_IPV4_ROUTE_ADD] 19
2022/07/06 14:36:56 ZEBRA: zebra message received [ZEBRA_IPV4_ROUTE_DELETE] 19
2022/07/06 14:36:56 ZEBRA: zebra message received [ZEBRA_IPV4_ROUTE_ADD] 19
2022/07/06 14:42:53 ZEBRA: zebra message received [ZEBRA_IPV4_ROUTE_DELETE] 19
2022/07/06 14:42:53 ZEBRA: zebra message received [ZEBRA_IPV4_ROUTE_DELETE] 19
2022/07/06 14:42:53 ZEBRA: zebra message received [ZEBRA_IPV4_ROUTE_DELETE] 19
2022/07/06 14:42:53 ZEBRA: zebra message received [ZEBRA_IPV4_ROUTE_DELETE] 19
```

۸ سوال ۷

با استفاده از دستور telnet localhost ripd وارد مد مانیتورینگ می‌شویم و از دستور show ip rip محتوای پایگاه داده RIP را در روتر r5 مشاهده می‌کنیم.

```
node: r5 (r5)
root@r5:/# telnet localhost ripd
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

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

User Access Verification

Password:
r5> show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
          (n) - normal, (s) - static, (d) - default, (r) - redistribute,
          (i) - interface

      Network        Next Hop        Metric From        Tag Time
R(n) 10.10.11.0/24    10.10.45.4          3 10.10.45.4        0 02:36
R(n) 10.10.12.0/24    10.10.25.2          2 10.10.25.2        0 02:54
R(n) 10.10.14.0/24    10.10.45.4          2 10.10.45.4        0 02:36
R(n) 10.10.23.0/24    10.10.35.3          2 10.10.35.3        0 02:38
C(i) 10.10.25.0/24    0.0.0.0             1 self              0
C(i) 10.10.35.0/24    0.0.0.0             1 self              0
C(i) 10.10.45.0/24    0.0.0.0             1 self              0
r5>
```

۹ سوال ۸

با توجه به دستور network 10.10.11.0/24 در پیکربندی RIP روتر r1 با توجه به اینکه رابط r1-eth3 را برای ارسال و دریافت بروزرسانی‌های RIP فعالسازی می‌کند، از سمت h1 بخاطر عدم اجرای RIP قابل قبول واقع نمی‌شود و رد می‌شود. در نتیجه تا زمانی که دستور مربوطه کامنت نشود، h1 نمی‌تواند به کمک دیتابیس r1 اطلاعات مسیریابی خود را بروزرسانی کند.

۱۰ سوال ۹

دستورهای ping و traceroute مربوطه را روی h1 اجرا می‌کنیم.

```
node: h1 (h1)
root@h1:/# traceroute 10.10.23.2
traceroute to 10.10.23.2 (10.10.23.2), 64 hops max
 1  10.10.11.1  7.212ms  3.724ms  4.919ms
 2  10.10.23.2  10.863ms  13.459ms  4.918ms
root@h1:/# ping 10.10.23.2
PING 10.10.23.2 (10.10.23.2) 56(84) bytes of data:
64 bytes from 10.10.23.2: icmp_seq=1 ttl=63 time=3.35 ms
64 bytes from 10.10.23.2: icmp_seq=2 ttl=63 time=0.602 ms
64 bytes from 10.10.23.2: icmp_seq=3 ttl=63 time=0.623 ms
64 bytes from 10.10.23.2: icmp_seq=4 ttl=63 time=0.572 ms
64 bytes from 10.10.23.2: icmp_seq=5 ttl=63 time=0.639 ms
64 bytes from 10.10.23.2: icmp_seq=6 ttl=63 time=0.570 ms
^C
--- 10.10.23.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5017ms
rtt min/avg/max/mdev = 0.570/1.059/3.351/1.025 ms
root@h1:/#
```

```
node: h1 (h1)
root@h1:/# traceroute 10.10.12.2
traceroute to 10.10.12.2 (10.10.12.2), 64 hops max
 1  10.10.11.1  0.983ms  3.088ms  1.743ms
 2  10.10.12.2  6.081ms  5.800ms  5.757ms
root@h1:/# ping 10.10.12.2
PING 10.10.12.2 (10.10.12.2) 56(84) bytes of data:
64 bytes from 10.10.12.2: icmp_seq=1 ttl=63 time=16.6 ms
64 bytes from 10.10.12.2: icmp_seq=2 ttl=63 time=2.45 ms
64 bytes from 10.10.12.2: icmp_seq=3 ttl=63 time=1.43 ms
64 bytes from 10.10.12.2: icmp_seq=4 ttl=63 time=0.594 ms
64 bytes from 10.10.12.2: icmp_seq=5 ttl=63 time=0.595 ms
^C
--- 10.10.12.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4016ms
rtt min/avg/max/mdev = 0.594/4.347/16.651/6.190 ms
root@h1:/#
```

۱۱ سوال ۱۰

دستور network مربوطه را کامنت می کنیم و اقدام به ریست کردن سرویس quagga می کنیم.

```
node: r1 (r1)
GNU nano 2.2.6 File: ripd.conf Modified
router rip
version 2

redistribute connected

network 10.10.11.0/24
!network 10.10.12.0/24
network 10.10.14.0/24

line vty

Get Help WriteOut Read File Prev Page Cut Text Cur Pos
Exit Justify Where Is Next Page UnCut Text To Spell
```

۱۲ سوال ۱۱

حال دوباره دستورهای ping و traceroute مربوطه را روی h1 اجرا می کنیم.

```
node: h1 (h1)
root@h1:~# traceroute 10.10.23.2
traceroute to 10.10.23.2 (10.10.23.2), 64 hops max
 1  10.10.11.1  4.385ms  3.776ms  3.812ms
 2  10.10.14.4  8.904ms  8.805ms  4.261ms
 3  10.10.45.5  20.311ms  59.231ms  62.474ms
 4  10.10.23.3  102.477ms  44.939ms  72.134ms
 5  10.10.23.2  111.710ms  69.250ms  79.737ms
root@h1:~# ping 10.10.23.2
PING 10.10.23.2 (10.10.23.2) 56(84) bytes of data:
64 bytes from 10.10.23.2: icmp_seq=1 ttl=63 time=18.5 ms
64 bytes from 10.10.23.2: icmp_seq=2 ttl=63 time=19.0 ms
64 bytes from 10.10.23.2: icmp_seq=3 ttl=63 time=1.37 ms
64 bytes from 10.10.23.2: icmp_seq=4 ttl=63 time=1.05 ms
64 bytes from 10.10.23.2: icmp_seq=5 ttl=63 time=1.02 ms
64 bytes from 10.10.23.2: icmp_seq=6 ttl=63 time=0.973 ms
^C
--- 10.10.23.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5020ms
rtt min/avg/max/ndev = 0.973/6.998/19.046/8.334 ms
root@h1:~#
```

```
node: h1 (h1)
root@h1:~# traceroute 10.10.12.2
traceroute to 10.10.12.2 (10.10.12.2), 64 hops max
 1  10.10.11.1  3.002ms  1.899ms  1.262ms
 2  10.10.12.2  11.979ms  1.073ms  6.132ms
root@h1:~# ping 10.10.12.2
PING 10.10.12.2 (10.10.12.2) 56(84) bytes of data:
64 bytes from 10.10.12.2: icmp_seq=1 ttl=63 time=20.1 ms
64 bytes from 10.10.12.2: icmp_seq=2 ttl=63 time=5.19 ms
64 bytes from 10.10.12.2: icmp_seq=3 ttl=63 time=0.588 ms
64 bytes from 10.10.12.2: icmp_seq=4 ttl=63 time=2.44 ms
64 bytes from 10.10.12.2: icmp_seq=5 ttl=63 time=1.66 ms
^C
--- 10.10.12.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4013ms
rtt min/avg/max/ndev = 0.588/6.017/20.190/7.248 ms
root@h1:~#
```

همانطور که مشاهده می شود، مقدار ping افزایش پیدا کرده است و در *traceroute* برای 10.10.23.2 تعداد ۵ تا انتری مشاهده می شود به نسبت قبل که دو تا انتری بود. دلیل آن این است که RIP دو تا روتر متفاوت اما با یک پیشوند یکسان 10.10.12.0/24 مشاهده می کند و این دو روتر هیچ ارتباطی با همدیگر ندارند. به همین علت، r2 مسیری که توسط روتر r5 به عنوان تنها راه رسیدن به r1 دارد را اشغال می کند و هم چنین برای جبران، شبکه های جدید 10.10.14.0/24 و 10.10.11.0/24 برای رسیدن به r5 نصب می شوند.

۱۳ قسمت ج

ابتدا فایل `lab8_network_v2.py` را برای ساخت شبکه اجرا می کنیم و اقدام به ریست کردن سرویس `quagga` روی روترهای `r1`، `r2` و `r4` می کنیم.

The image shows three terminal windows, each representing a different router in a network simulation. The windows are titled 'node: r4 (r4)', 'node: r2 (r2)', and 'node: r1 (r1)'. Each window displays the commands used to stop and start the Quagga service. The 'stop' command lists the daemons to be stopped: ripd, zebra, bgpd, ripngd, ospfd, ospf6d, isisd, and babeld. The 'start' command lists the daemons to be started. The output shows that the services were successfully stopped and started on all three routers.

```
node: r4 (r4)
root@r4:~# /etc/init.d/quagga stop
Stopping Quagga monitor daemon: (watchquagga).
Stopping Quagga daemons (prio:0): ripd zebra (bgpd) (ripngd) (ospfd) (ospf6d) (isisd) (babeld).
Removing all routes made by zebra.
root@r4:~# /etc/init.d/quagga start
Loading capability module if not yet done.
Starting Quagga daemons (prio:0): zebra can't open logfile /home/mininet/Downloads/lab8/configs/r4/logs/zebra.log
ripd can't open logfile /home/mininet/Downloads/lab8/configs/r4/logs/ripd.log
root@r4:~#

node: r2 (r2)
root@r2:~# /etc/init.d/quagga stop
Stopping Quagga monitor daemon: (watchquagga).
Stopping Quagga daemons (prio:0): ripd zebra (bgpd) (ripngd) (ospfd) (ospf6d) (isisd) (babeld).
Removing all routes made by zebra.
root@r2:~# /etc/init.d/quagga start
Loading capability module if not yet done.
Starting Quagga daemons (prio:0): zebra can't open logfile /home/mininet/Downloads/lab8/configs/r2/logs/zebra.log
ripd can't open logfile /home/mininet/Downloads/lab8/configs/r2/logs/ripd.log
root@r2:~#

node: r1 (r1)
root@r1:~# /etc/init.d/quagga start
Loading capability module if not yet done.
Starting Quagga daemons (prio:0): zebra/usr/lib/quagga/zebra already running.
root@r1:~# /etc/init.d/quagga stop
Stopping Quagga monitor daemon: (watchquagga).
Stopping Quagga daemons (prio:0): ripd zebra (bgpd) (ripngd) (ospfd) (ospf6d) (isisd) (babeld).
Removing all routes made by zebra.
root@r1:~#
```

با استفاده از دستور `show ip rip` محتویات جدول مسیریابی `RIP` در روتر `r1` را می بینیم.

```
r1> show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
      (n) - normal, (s) - static, (d) - default, (r) - redistribute,
      (i) - interface
```

| | Network | Next Hop | Metric | From | Tag |
|------|---------------|------------|--------|------------|-----|
| C(i) | 10.10.11.0/24 | 0.0.0.0 | 1 | self | 0 |
| C(i) | 10.10.12.0/24 | 0.0.0.0 | 1 | self | 0 |
| C(i) | 10.10.14.0/24 | 0.0.0.0 | 1 | self | 0 |
| R(n) | 10.10.23.0/24 | 10.10.12.2 | 2 | 10.10.12.2 | 0 |
| R(n) | 10.10.25.0/24 | 10.10.12.2 | 2 | 10.10.12.2 | 0 |
| R(n) | 10.10.35.0/24 | 10.10.12.2 | 3 | 10.10.12.2 | 0 |
| R(n) | 10.10.45.0/24 | 10.10.14.4 | 2 | 10.10.14.4 | 0 |

سپس اقدام به *shutdown* کردن اینترفیس *r2-eth1* از روتر *r2* می‌کنیم.

```
node: r2 (r2)
root@r2:/# 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 Kunihiro Ishiguro, et al.

User Access Verification

Password:
r2> enable
Password:
r2# configure terminal
r2(config)# interface r2-eth1
r2(config-if)# shutdown
r2(config-if)#
```

حال مجدداً با استفاده از دستور *show ip rip* محتویات جدول مسیریابی *RIP* در روتر *r1* را می‌بینیم.

```
r1> show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
      (n) - normal, (s) - static, (d) - default, (r) - redistribute,
      (i) - interface

      Network          Next Hop          Metric From          Tag
C(i) 10.10.11.0/24      0.0.0.0           1 self               0
C(i) 10.10.12.0/24      0.0.0.0           1 self               0
C(i) 10.10.14.0/24      0.0.0.0           1 self               0
R(n) 10.10.23.0/24      10.10.14.4        4 10.10.14.4          0
R(n) 10.10.25.0/24      10.10.14.4        3 10.10.14.4          0
R(n) 10.10.35.0/24      10.10.14.4        3 10.10.14.4          0
R(n) 10.10.45.0/24      10.10.14.4        2 10.10.14.4          0
```

۱۴ سوال ۱۲

تغییراتی که رخ داده است این است که پس از انقضای تایمر نامعتبر ۱۸۰ ثانیه، مسیرهای *r2* از جدول مسیریابی *r1* حذف می‌شوند و با مسیرهای *r4* جایگزین می‌شوند. این کار پس از دریافت اولین بسته *hello* از *r4* به دنبال انقضای تایمر نامعتبر رخ می‌دهد.

۱۵ سوال ۱۳

پس از ۵ دقیقه اینترفیس *r2-eth1* را روی *r2* با دستور *no shutdown* بالا می‌آوریم.
سپس دوباره جدول مسیریابی را بررسی می‌کنیم.

```
r1> show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
      (n) - normal, (s) - static, (d) - default, (r) - redistribute,
      (i) - interface

      Network          Next Hop          Metric From      Tag
C(i) 10.10.11.0/24     0.0.0.0           1 self           0
C(i) 10.10.12.0/24     0.0.0.0           1 self           0
C(i) 10.10.14.0/24     0.0.0.0           1 self           0
R(n) 10.10.23.0/24     10.10.12.2        2 10.10.12.2     0
R(n) 10.10.25.0/24     10.10.12.2        2 10.10.12.2     0
R(n) 10.10.35.0/24     10.10.14.4        3 10.10.14.4     0
R(n) 10.10.45.0/24     10.10.14.4        2 10.10.14.4     0
r1>
```

همانطور که مشاهده می‌شود، محتوای جدول مانند اول نخواهد بود. دلیل آن این است که پس از نمایش مجدد رابط *r2-eth1* در *r2*، جدول مسیریابی *r1* مدتی طول می‌کشد تا همگرا شود و محتوای آن مانند ابتدا نخواهد بود. اما مسیرهای هزینه مساوی از طریق *r4* حفظ می‌شوند. (فقط مسیرهای برتر انتخاب می‌شوند.)

۱۶ قسمت د

ابتدا جدول مسیریابی *r2* را می‌بینیم.

```
r2> show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
      (n) - normal, (s) - static, (d) - default, (r) - redistribute,
      (i) - interface

      Network          Next Hop          Metric From      Tag
R(n) 10.10.11.0/24     10.10.12.1        2 10.10.12.1     0
C(i) 10.10.12.0/24     0.0.0.0           1 self           0
R(n) 10.10.14.0/24     10.10.12.1        2 10.10.12.1     0
C(i) 10.10.23.0/24     0.0.0.0           1 self           0
C(i) 10.10.25.0/24     0.0.0.0           1 self           0
R(n) 10.10.35.0/24     10.10.23.3        2 10.10.23.3     0
R(n) 10.10.45.0/24     10.10.25.5        2 10.10.25.5     0
r2> enable
```

سپس با اجرای دستورات زیر در مد *config* روتر دوباره جدول مسیریابی را می بینیم.

Offset-list addExtraMetric in 4 r2-eth1

Access-list addExtraMetric permit any

```
r2> enable
Password:
r2# configure terminal
r2(config)# router rip
r2(config-router)# offset-list addExtraMetric in 4 r2-eth1
r2(config-router)# access-list addExtraMetric permit any
r2(config)# exit
r2# show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
      (n) - normal, (s) - static, (d) - default, (r) - redistribute,
      (i) - interface
```

| | Network | Next Hop | Metric From | Tag |
|------|---------------|------------|--------------|-----|
| R(n) | 10.10.11.0/24 | 10.10.25.5 | 4 10.10.25.5 | 0 |
| C(i) | 10.10.12.0/24 | 0.0.0.0 | 1 self | 0 |
| R(n) | 10.10.14.0/24 | 10.10.25.5 | 3 10.10.25.5 | 0 |
| C(i) | 10.10.23.0/24 | 0.0.0.0 | 1 self | 0 |
| C(i) | 10.10.25.0/24 | 0.0.0.0 | 1 self | 0 |
| R(n) | 10.10.35.0/24 | 10.10.23.3 | 2 10.10.23.3 | 0 |
| R(n) | 10.10.45.0/24 | 10.10.25.5 | 2 10.10.25.5 | 0 |

دلیل آن این است که از *r2* ، می توانیم به طور مستقیم یک مسیر برای محاسبه هزینه ی *r2* بدهیم. در این صورت *r2* بروزرسانی ها را از *r1* دریافت می کند و برای محاسبه کوتاه ترین درخت مسیر، آفست مورد نظر را اضافه می کند.

حال جدول مسیریابی *r1* و *r4* را ثبت می کنیم.

```
r1> show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
      (n) - normal, (s) - static, (d) - default, (r) - redistribute,
      (i) - interface
```

| | Network | Next Hop | Metric From | Tag |
|------|---------------|------------|--------------|-----|
| C(i) | 10.10.11.0/24 | 0.0.0.0 | 1 self | 0 |
| C(i) | 10.10.12.0/24 | 0.0.0.0 | 1 self | 0 |
| C(i) | 10.10.14.0/24 | 0.0.0.0 | 1 self | 0 |
| R(n) | 10.10.23.0/24 | 10.10.12.2 | 2 10.10.12.2 | 0 |
| R(n) | 10.10.25.0/24 | 10.10.12.2 | 2 10.10.12.2 | 0 |
| R(n) | 10.10.35.0/24 | 10.10.12.2 | 3 10.10.12.2 | 0 |
| R(n) | 10.10.45.0/24 | 10.10.14.4 | 2 10.10.14.4 | 0 |

```
r4> show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
      (n) - normal, (s) - static, (d) - default, (r) - redistribute,
      (i) - interface
```

| | Network | Next Hop | Metric From | Tag |
|------|---------------|------------|--------------|-----|
| R(n) | 10.10.11.0/24 | 10.10.14.1 | 2 10.10.14.1 | 0 |
| R(n) | 10.10.12.0/24 | 10.10.14.1 | 2 10.10.14.1 | 0 |
| C(i) | 10.10.14.0/24 | 0.0.0.0 | 1 self | 0 |
| R(n) | 10.10.23.0/24 | 10.10.14.1 | 3 10.10.14.1 | 0 |
| R(n) | 10.10.25.0/24 | 10.10.45.5 | 2 10.10.45.5 | 0 |
| R(n) | 10.10.35.0/24 | 10.10.45.5 | 2 10.10.45.5 | 0 |
| C(i) | 10.10.45.0/24 | 0.0.0.0 | 1 self | 0 |

در ادامه اقدام به نابود کردن پروسس های *ripd* در روترهای ۲ و ۵ می کنیم.

در نهایت، دوباره جداول مسیریابی $r1$ و $r4$ را ثبت می‌کنیم.

| <pre>r1> show ip rip Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP Sub-codes: (n) - normal, (s) - static, (d) - default, (r) - redistribute, (i) - interface</pre> | | | | | <pre>r4> show ip rip Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP Sub-codes: (n) - normal, (s) - static, (d) - default, (r) - redistribute, (i) - interface</pre> | | | | |
|--|------------|---------------|-----|--|--|------------|---------------|-----|--|
| Network | Next Hop | Metric From | Tag | | Network | Next Hop | Metric From | Tag | |
| C(i) 10.10.11.0/24 | 0.0.0.0 | 1 self | 0 | | R(n) 10.10.11.0/24 | 10.10.14.1 | 2 10.10.14.1 | 0 | |
| C(i) 10.10.12.0/24 | 0.0.0.0 | 1 self | 0 | | R(n) 10.10.12.0/24 | 10.10.14.1 | 2 10.10.14.1 | 0 | |
| C(i) 10.10.14.0/24 | 0.0.0.0 | 1 self | 0 | | C(i) 10.10.14.0/24 | 0.0.0.0 | 1 self | 0 | |
| R(n) 10.10.23.0/24 | 10.10.12.2 | 16 10.10.12.2 | 0 | | R(n) 10.10.23.0/24 | 10.10.14.1 | 16 10.10.14.1 | 0 | |
| R(n) 10.10.25.0/24 | 10.10.12.2 | 16 10.10.12.2 | 0 | | R(n) 10.10.25.0/24 | 10.10.45.5 | 16 10.10.45.5 | 0 | |
| R(n) 10.10.35.0/24 | 10.10.12.2 | 16 10.10.12.2 | 0 | | R(n) 10.10.35.0/24 | 10.10.45.5 | 16 10.10.45.5 | 0 | |
| R(n) 10.10.45.0/24 | 10.10.14.4 | 2 10.10.14.4 | 0 | | C(i) 10.10.45.0/24 | 0.0.0.0 | 1 self | 0 | |

همانطور که پیش‌تر دیدیم، نتیجه‌ی جدول فوق تغییر می‌کند و سابنت‌هایی که مستقیماً به $r3$ متصل هستند، نیز اضافه شده‌اند. دلیل آن این است که باید مقداری صبر کنیم تا این سابنت‌ها از این دو جدول حذف شوند. برای این کار حدود ۳ تا ۴ دقیقه زمان لازم داریم تا همگرایی صورت بگیرد.