CSE 534 – Assignment 3

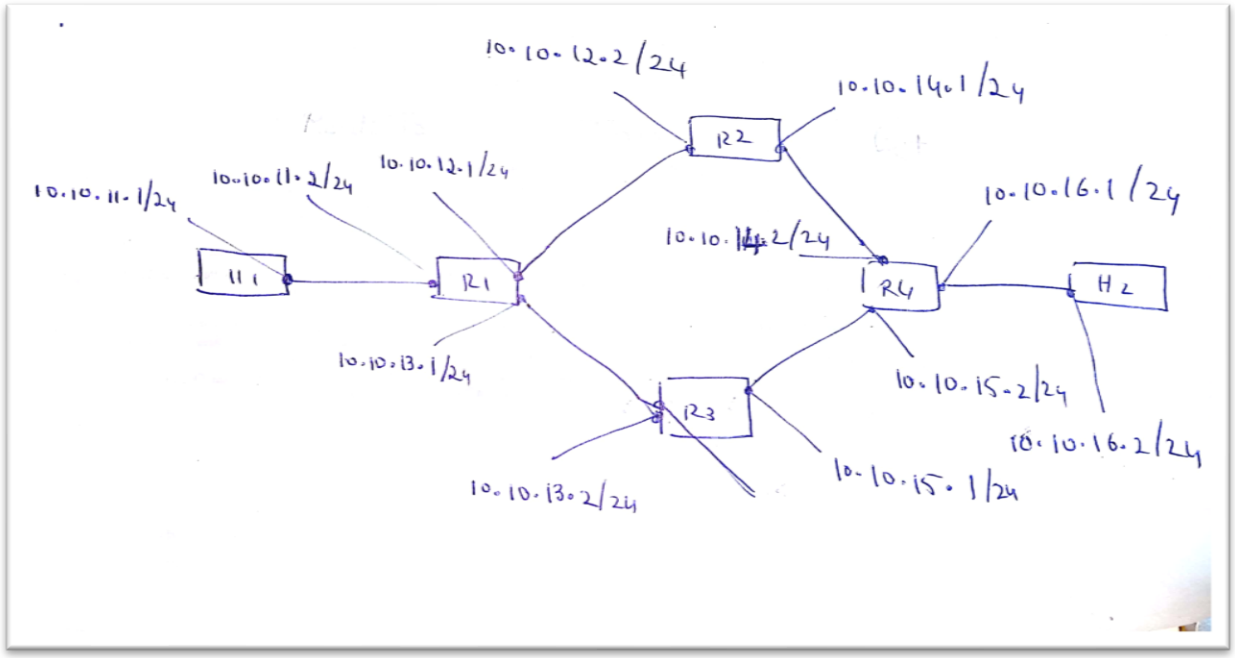
Adarsh Alangar, 112026947

Part A.

Part A.1. Creating the network topology

a) File included as topo.py

b) All subnets masks are "/24" in the below topology



The subnets are as follows:

Subnet	Nodes
10.10.11.0/24	H1, R1
10.10.12.0/24	R1, R2
10.10.13.0/24	R1, R3
10.10.14.0/24	R2, R4
10.10.15.0/24	R3, R4
10.10.16.0/24	R4, H2

Part A.2. Configuring h1 to ping h2

a)

Configuration: Modified the file "/etc/sysctl.conf". Changed the "net.ipv4.ip_forward" bit to 1 from a 0. To enable the change, "sysctl -p /etc/sysctl.conf" is run. This must be done once at the mininet

level and it's automatically applied to all routers on the next mininext restart.
The following commands are then run to enable just the minimum static routing required to get h1 to ping h2

Route from h1 to h2

```
h1 ip route add default via 10.10.11.2 dev h1-eth0
r1 ip route add 10.10.16.0/24 via 10.10.12.2 dev r1-eth1
r2 ip route add 10.10.16.0/24 via 10.10.14.2 dev r2-eth1
```

Route from h2 to h1

```
h2 ip route add default via 10.10.16.1 dev h2-eth0
r4 ip route add 10.10.11.0/24 via 10.10.14.1 dev r4-eth0
r2 ip route add 10.10.11.0/24 via 10.10.12.1 dev r2-eth0
```

Forward Path:

- For h1 we add a default entry to the (default gateway) router r1.
- At r1 and r2 we provide a path to the subnet 10.10.16.0/24, so that they forward all packets of this range to r2 and r4 respectively.
- r4 doesn't need to be configured as it has a direct link.

Reverse Path:

- For h2 we add a default entry to the (default gateway) router r4.
- At r4 and r2 we provide a path to the subnet 10.10.11.0/24, so that they forward all packets of this range to r2 and r1 respectively.
- r1 doesn't need to be configured as it has a direct link.

Routing Tables

H1

```
mininext> h1 ip route
default via 10.10.11.2 dev h1-eth0
10.10.11.0/24 dev h1-eth0  proto kernel  scope link  src 10.10.11.1
```

R1

```
mininext> r1 ip route
10.10.11.0/24 dev r1-eth0  proto kernel  scope link  src 10.10.11.2
10.10.12.0/24 dev r1-eth1  proto kernel  scope link  src 10.10.12.1
10.10.13.0/24 dev r1-eth2  proto kernel  scope link  src 10.10.13.1
10.10.16.0/24 via 10.10.12.2 dev r1-eth1
mininext>
```

R2

```
mininext> r2 ip route
10.10.11.0/24 via 10.10.12.1 dev r2-eth0
10.10.12.0/24 dev r2-eth0  proto kernel  scope link  src 10.10.12.2
10.10.14.0/24 dev r2-eth1  proto kernel  scope link  src 10.10.14.1
10.10.16.0/24 via 10.10.14.2 dev r2-eth1
mininext>
```

R3

```
mininext> r3 ip route
10.10.13.0/24 dev r3-eth0  proto kernel  scope link  src 10.10.13.2
10.10.15.0/24 dev r3-eth1  proto kernel  scope link  src 10.10.15.1
mininext>
```

R4

```
mininext> r4 ip route
10.10.11.0/24 via 10.10.14.1 dev r4-eth0
10.10.14.0/24 dev r4-eth0  proto kernel  scope link  src 10.10.14.2
10.10.15.0/24 dev r4-eth2  proto kernel  scope link  src 10.10.15.2
10.10.16.0/24 dev r4-eth1  proto kernel  scope link  src 10.10.16.1
mininext>
```

H2

```
mininext> h2 ip route
default via 10.10.16.1 dev h2-eth0
10.10.11.0/24 via 10.10.16.1 dev h2-eth0
10.10.16.0/24 dev h2-eth0  proto kernel  scope link  src 10.10.16.2
mininext>
```

b) Traceroute output

To enable traceroute, run:

```
sudo apt-get install traceroute
```

This enable it on all nodes. The output is below:

```
mininext> h1 traceroute h2
traceroute to 10.10.16.2 (10.10.16.2), 30 hops max, 60 byte packets
 1  10.10.11.2 (10.10.11.2)  0.025 ms  0.008 ms  0.014 ms
 2  10.10.12.2 (10.10.12.2)  0.015 ms  0.010 ms  0.009 ms
 3  * * *
 4  10.10.16.2 (10.10.16.2)  0.029 ms  0.016 ms  0.016 ms
mininext>
```

Part B.

Part B.1. Configuring RIP

a) & b) Commands and explanation

- **Enabling daemons:** In mininet, the each node h1,h2, r1-r4 has a separate folder under ~/…./examples/quagga-ixp/configs/. Here the **daemons file** is modified to have **zebra** and **ripd** configured to **yes** for all node.
- Create two config files **zebra.conf** (empty, required by ripd to run), and **ripd.conf** (RIP daemon configuration).
- In the ripd conf file, the interfaces of the node are defined (sample included in zip). This enables rip for the defined interfaces.
- In mininet, change the owner and group of config files to quagga and quaggavty respectively. Also, change file permissions to 640. Without the following two commands, ripd does not start:
  ```
  chown quagga.quaggavty /etc/quagga/*.conf
  sudo chmod 640 /etc/quagga/*.conf
  ```
- Start the quagga daemon by running:
  ```
  /etc/init.d/quagga start
  ```
- These config files are copied on start-by by mininext to the **/etc/quagga/** for each node. ripd then works by default without additional configuration.

Part B.2. Running RIP

a) Routing table at each node

r1:

```
r1# sshhooww  iipp  rroouuttee

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-eth0
C>* 10.10.12.0/24 is directly connected, r1-eth1
C>* 10.10.13.0/24 is directly connected, r1-eth2
R>* 10.10.14.0/24 [120/2] via 10.10.12.2, r1-eth1, 00:02:07
R>* 10.10.15.0/24 [120/2] via 10.10.13.2, r1-eth2, 00:02:07
R>* 10.10.16.0/24 [120/3] via 10.10.12.2, r1-eth1, 00:02:07
C>* 127.0.0.0/8 is directly connected, lo
r1#
```

```
mininext> r1 ip route
10.10.11.0/24 dev r1-eth0  proto kernel  scope link  src 10.10.11.2
10.10.12.0/24 dev r1-eth1  proto kernel  scope link  src 10.10.12.1
10.10.13.0/24 dev r1-eth2  proto kernel  scope link  src 10.10.13.1
10.10.14.0/24 via 10.10.12.2 dev r1-eth1  proto zebra  metric 2
10.10.15.0/24 via 10.10.13.2 dev r1-eth2  proto zebra  metric 2
10.10.16.0/24 via 10.10.12.2 dev r1-eth1  proto zebra  metric 3
mininext>
```

r2:

```
r2# sshhooww  iipp  rroouuttee

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

R>* 10.10.11.0/24 [120/2] via 10.10.12.1, r2-eth0, 00:03:49
C>* 10.10.12.0/24 is directly connected, r2-eth0
R>* 10.10.13.0/24 [120/2] via 10.10.12.1, r2-eth0, 00:03:49
C>* 10.10.14.0/24 is directly connected, r2-eth1
R>* 10.10.15.0/24 [120/2] via 10.10.14.2, r2-eth1, 00:03:48
R>* 10.10.16.0/24 [120/2] via 10.10.14.2, r2-eth1, 00:03:48
C>* 127.0.0.0/8 is directly connected, lo
r2#
```

```
mininext> r2  ip route
10.10.11.0/24 via 10.10.12.1 dev r2-eth0  proto zebra  metric 2
10.10.12.0/24 dev r2-eth0  proto kernel  scope link  src 10.10.12.2
10.10.13.0/24 via 10.10.12.1 dev r2-eth0  proto zebra  metric 2
10.10.14.0/24 dev r2-eth1  proto kernel  scope link  src 10.10.14.1
10.10.15.0/24 via 10.10.14.2 dev r2-eth1  proto zebra  metric 2
10.10.16.0/24 via 10.10.14.2 dev r2-eth1  proto zebra  metric 2
mininext>
```

r3:

```
r3# sshhooww  iipp  rroouuttee

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

R>* 10.10.11.0/24 [120/2] via 10.10.13.1, r3-eth0, 00:05:16
R>* 10.10.12.0/24 [120/2] via 10.10.13.1, r3-eth0, 00:05:16
C>* 10.10.13.0/24 is directly connected, r3-eth0
R>* 10.10.14.0/24 [120/2] via 10.10.15.2, r3-eth1, 00:05:15
C>* 10.10.15.0/24 is directly connected, r3-eth1
R>* 10.10.16.0/24 [120/2] via 10.10.15.2, r3-eth1, 00:05:15
C>* 127.0.0.0/8 is directly connected, lo
r3#
```

```
mininext> r3 ip route
10.10.11.0/24 via 10.10.13.1 dev r3-eth0  proto zebra  metric 2
10.10.12.0/24 via 10.10.13.1 dev r3-eth0  proto zebra  metric 2
10.10.13.0/24 dev r3-eth0  proto kernel  scope link  src 10.10.13.2
10.10.14.0/24 via 10.10.15.2 dev r3-eth1  proto zebra  metric 2
10.10.15.0/24 dev r3-eth1  proto kernel  scope link  src 10.10.15.1
10.10.16.0/24 via 10.10.15.2 dev r3-eth1  proto zebra  metric 2
mininext>
```

r4:

```
r4# sshhooww  iipp  rroouuttee

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

R>* 10.10.11.0/24 [120/3] via 10.10.14.1, r4-eth0, 00:06:17
R>* 10.10.12.0/24 [120/2] via 10.10.14.1, r4-eth0, 00:06:17
R>* 10.10.13.0/24 [120/2] via 10.10.15.1, r4-eth2, 00:06:17
C>* 10.10.14.0/24 is directly connected, r4-eth0
C>* 10.10.15.0/24 is directly connected, r4-eth2
C>* 10.10.16.0/24 is directly connected, r4-eth1
C>* 127.0.0.0/8 is directly connected, lo
r4#
```

```
mininext> r4 ip route
10.10.11.0/24 via 10.10.14.1 dev r4-eth0  proto zebra  metric 3
10.10.12.0/24 via 10.10.14.1 dev r4-eth0  proto zebra  metric 2
10.10.13.0/24 via 10.10.15.1 dev r4-eth2  proto zebra  metric 2
10.10.14.0/24 dev r4-eth0  proto kernel  scope link  src 10.10.14.2
10.10.15.0/24 dev r4-eth2  proto kernel  scope link  src 10.10.15.2
10.10.16.0/24 dev r4-eth1  proto kernel  scope link  src 10.10.16.1
mininext>
```

h1:

```
mininext> h1 ip route
10.10.11.0/24 dev h1-eth0  proto kernel  scope link  src 10.10.11.1
10.10.12.0/24 via 10.10.11.2 dev h1-eth0  proto zebra  metric 2
10.10.13.0/24 via 10.10.11.2 dev h1-eth0  proto zebra  metric 2
10.10.14.0/24 via 10.10.11.2 dev h1-eth0  proto zebra  metric 3
10.10.15.0/24 via 10.10.11.2 dev h1-eth0  proto zebra  metric 3
10.10.16.0/24 via 10.10.11.2 dev h1-eth0  proto zebra  metric 4
mininext>
```

```
h1# sshhooww  iipp  rroouuttee

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, h1-eth0
R>* 10.10.12.0/24 [120/2] via 10.10.11.2, h1-eth0, 00:01:52
R>* 10.10.13.0/24 [120/2] via 10.10.11.2, h1-eth0, 00:01:52
R>* 10.10.14.0/24 [120/3] via 10.10.11.2, h1-eth0, 00:01:51
R>* 10.10.15.0/24 [120/3] via 10.10.11.2, h1-eth0, 00:01:51
R>* 10.10.16.0/24 [120/4] via 10.10.11.2, h1-eth0, 00:01:51
C>* 127.0.0.0/8 is directly connected, lo
h1#
```

h2:

```
mininext> h2 ip route
10.10.11.0/24 via 10.10.16.1 dev h2-eth0  proto zebra  metric 4
10.10.12.0/24 via 10.10.16.1 dev h2-eth0  proto zebra  metric 3
10.10.13.0/24 via 10.10.16.1 dev h2-eth0  proto zebra  metric 3
10.10.14.0/24 via 10.10.16.1 dev h2-eth0  proto zebra  metric 2
10.10.15.0/24 via 10.10.16.1 dev h2-eth0  proto zebra  metric 2
10.10.16.0/24 dev h2-eth0  proto kernel  scope link  src 10.10.16.2
mininext>
```

```
h2# sshhooww  iipp  rroouuttee

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

R>* 10.10.11.0/24 [120/4] via 10.10.16.1, h2-eth0, 00:01:12
R>* 10.10.12.0/24 [120/3] via 10.10.16.1, h2-eth0, 00:01:12
R>* 10.10.13.0/24 [120/3] via 10.10.16.1, h2-eth0, 00:01:12
R>* 10.10.14.0/24 [120/2] via 10.10.16.1, h2-eth0, 00:01:12
R>* 10.10.15.0/24 [120/2] via 10.10.16.1, h2-eth0, 00:01:12
C>* 10.10.16.0/24 is directly connected, h2-eth0
C>* 127.0.0.0/8 is directly connected, lo
h2#
```

b) Traceroute h1 to h2

```
mininext> h1 traceroute h2
traceroute to 10.10.16.2 (10.10.16.2), 30 hops max, 60 byte packets
 1  10.10.11.2 (10.10.11.2)  0.022 ms  0.033 ms  0.006 ms
 2  10.10.12.2 (10.10.12.2)  0.012 ms  0.008 ms  0.007 ms
 3  10.10.14.2 (10.10.14.2)  0.015 ms  0.010 ms  0.009 ms
 4  10.10.16.2 (10.10.16.2)  0.016 ms  0.012 ms  0.011 ms
mininext>
```

c) Time to ping:

Time to ping: 8996ms on average

```
mininext> h1 ping -D h2
PING 10.10.16.2 (10.10.16.2) 56(84) bytes of data.
[1542078984.530412] 64 bytes from 10.10.16.2: icmp_seq=1 ttl=61 time=0.037 ms
[1542078985.529427] 64 bytes from 10.10.16.2: icmp_seq=2 ttl=61 time=0.047 ms
[1542078986.528437] 64 bytes from 10.10.16.2: icmp_seq=3 ttl=61 time=0.052 ms
[1542078987.527451] 64 bytes from 10.10.16.2: icmp_seq=4 ttl=61 time=0.044 ms
[1542078988.526917] 64 bytes from 10.10.16.2: icmp_seq=5 ttl=61 time=0.046 ms
[1542078989.527137] 64 bytes from 10.10.16.2: icmp_seq=6 ttl=61 time=0.053 ms
[1542078990.527594] 64 bytes from 10.10.16.2: icmp_seq=7 ttl=61 time=0.040 ms
[1542078991.526928] 64 bytes from 10.10.16.2: icmp_seq=8 ttl=61 time=0.048 ms
[1542078992.527701] 64 bytes from 10.10.16.2: icmp_seq=9 ttl=61 time=0.043 ms
[1542078993.527048] 64 bytes from 10.10.16.2: icmp_seq=10 ttl=61 time=0.072 ms
^C
--- 10.10.16.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 8996ms
rtt min/avg/max/mdev = 0.037/0.048/0.072/0.010 ms
mininext>
```

d) Convergence time:

To estimate the convergence, we send a "SIGHUP" (kill -1) to any router (here r4). This instructs the ripd to clear all routes in it's routing table. Therefore, we send a sighup and then start out measurement of time till ping. Commands:

- `r4 kill -1 \`cat /var/run/quagga/ripd.pid\` && date +%s.%N`
- `h1 ping -aD h2`

The above commands does the following:

- Clear r1's routing table
- Print the current timestamp
- Ping h2 from h1 repeatedly (-a) with a timestamp (-D)

- **Convergence time**: **~17.5s**

```
mininext> r4 kill -1 `cat /var/run/quagga/ripd.pid` && date +%s.%N
1542270770.435775649
mininext> h1 ping -aD h2
PING 10.10.16.2 (10.10.16.2) 56(84) bytes of data.
[1542270787.921890] 64 bytes from 10.10.16.2: icmp_seq=17 ttl=61 time=0.096 ms
[1542270788.923236] 64 bytes from 10.10.16.2: icmp_seq=18 ttl=61 time=0.055 ms
[1542270789.923410] 64 bytes from 10.10.16.2: icmp_seq=19 ttl=61 time=0.115 ms
[1542270790.926017] 64 bytes from 10.10.16.2: icmp_seq=20 ttl=61 time=0.136 ms
[1542270791.928183] 64 bytes from 10.10.16.2: icmp_seq=21 ttl=61 time=0.105 ms
[1542270792.930067] 64 bytes from 10.10.16.2: icmp_seq=22 ttl=61 time=0.096 ms
^C
--- 10.10.16.2 ping statistics ---
22 packets transmitted, 6 received, 72% packet loss, time 21136ms
rtt min/avg/max/mdev = 0.055/0.100/0.136/0.026 ms
mininext>
```

Part B.3. Bringing down links

a) **Getting the link down:** Mininet provides a convenient command: link up/down. This is used to simulate a link failure or undo it. This was used as follows:

```
link r1 r2 down
```

b) Time for connectivity to get backup: ~5s

```
mininext> link r1 r2 down
mininext> h1 date +%s.%N && ping -aD h2
1542081906.511136429
PING 10.10.16.2 (10.10.16.2) 56(84) bytes of data.
[1542081906.513047] From 10.10.11.2 icmp_seq=1 Destination Net Unreachable
[1542081907.512059] From 10.10.11.2 icmp_seq=2 Destination Net Unreachable
[1542081908.511041] From 10.10.11.2 icmp_seq=3 Destination Net Unreachable
[1542081909.510054] From 10.10.11.2 icmp_seq=4 Destination Net Unreachable
[1542081911.510458] 64 bytes from 10.10.11.2: icmp_seq=6 ttl=61 time=0.045 ms
[1542081912.509941] 64 bytes from 10.10.11.2: icmp_seq=7 ttl=61 time=0.044 ms
[1542081913.510260] 64 bytes from 10.10.11.2: icmp_seq=8 ttl=61 time=0.048 ms
[1542081914.509953] 64 bytes from 10.10.11.2: icmp_seq=9 ttl=61 time=0.052 ms
```

c) Traceroute of the new path

```
mininext> h1 traceroute h2
traceroute to 10.10.16.2 (10.10.16.2), 30 hops max, 60 byte packets
 1  10.10.11.2 (10.10.11.2)  0.024 ms  0.011 ms  0.007 ms
 2  10.10.13.2 (10.10.13.2)  0.017 ms  0.011 ms  0.009 ms
 3  10.10.15.2 (10.10.15.2)  0.021 ms  0.013 ms  0.013 ms
 4  10.10.16.2 (10.10.16.2)  0.019 ms  0.015 ms  0.015 ms
mininext>
```

Part C: RIP Lite

Part C.1.

a) Code attached in zip. The has a dependency on the following python modules apart from the regular ones:

- tabulate – For printing tables
- threading
- socket

Running the code:

- The main file with the code is called **bfd.py.** To run, it required two other files that must be shipped together
  - **config.py** (holds constants and simple configurations for the script)
  - **bf.json** (contains costs, neighbours and interfaces of the node)
- To deploy it, create a bf.json file in each node and place it with the config.py and bfd.py file

- The script runs on two threads
    - One to handle messages
    - One to listen for new incoming messages
- It is recommended to run the code as below:
  `r1 python -u bfd.py >log.txt 2>&1 &`
- This create an unbuffered output (-u) to a log.txt file. The last '&' runs the process in the background.
- Two output files are used by the script
    - **log.txt**: Most logs are written to this file. The routing table is visible here.
    - **raw_send_rec_log.txt**: Contains all received messages to the script. This is used by the listener thread as its output.

b) **Time to find the shortest path**

We start node h2 after all other nodes. As soon as the node is up, it logs the time. Similarly, h1 logs the timestamp of the update of its routing table when it got the optimal path to h2. We compare the difference between these two times to get the time taken to find the shortest path.

The time difference is: **~1.639s**

```
mininext> h2 cat log.txt
Configuration read
Spawning a thread
('Sending data to: ', u'r4', '{"id": "h2", "10.10.16.1": ["r4", 2], "10.10.16.2": ["h2", 0]}END')
('Server up at: ', '2018-11-15 05:52:27.181')
```

```
2018-11-15 05:52:28.762
Destination Network    Next Hop    Next Hop IP      Cost
---------------------  ----------  -------------  ------
10.10.11.2             Direct      -                   2
10.10.14.1             r1          10.10.11.2         12
10.10.14.2             r1          10.10.11.2         16
10.10.12.2             r1          10.10.11.2         12
10.10.12.1             r1          10.10.11.2          2
10.10.13.2             r1          10.10.11.2          8
10.10.16.2             r1          10.10.11.2         18
10.10.16.1             r1          10.10.11.2         16
10.10.13.1             r1          10.10.11.2          2
10.10.15.1             r1          10.10.11.2          8
10.10.15.2             r1          10.10.11.2         13
('Sending data to: ', u'r1', '{"10.10.14.1": ["r1", 12], "10.10.11.1"
0.10.13.2": ["r1", 8], "10.10.16.2": ["r1", 18], "10.10.16.1": ["r1",
('Received: ', '{"10.10.14.1": ["r2", 10], "10.10.11.1": ["h1", 2], "
r3", 6], "10.10.16.2": ["r3", 13], "10.10.16.1": ["r3", 11], "10.10.1
{u'r1': u'10.10.11.2'}
The DV was changed. So we send the updated costs to everyone
2018-11-15 05:52:28.820
Destination Network    Next Hop    Next Hop IP      Cost
---------------------  ----------  -------------  ------
10.10.11.2             Direct      -                   2
10.10.14.1             r1          10.10.11.2         12
10.10.14.2             r1          10.10.11.2         13
10.10.12.2             r1          10.10.11.2         12
10.10.12.1             r1          10.10.11.2          2
10.10.13.2             r1          10.10.11.2          8
10.10.16.2             r1          10.10.11.2         15
10.10.16.1             r1          10.10.11.2         13
10.10.13.1             r1          10.10.11.2          2
10.10.15.1             r1          10.10.11.2          8
10.10.15.2             r1          10.10.11.2         13
```

c) Routing table at each node

h1

```
2018-11-15 09:43:08.094
Destination Network    Next Hop    Next Hop IP      Cost
--------------------  ----------  ------------  ------
10.10.11.2             Direct      -                 2
10.10.14.1             r1          10.10.11.2       12
10.10.14.2             r1          10.10.11.2       13
10.10.12.2             r1          10.10.11.2       12
10.10.12.1             r1          10.10.11.2        2
10.10.13.2             r1          10.10.11.2        8
10.10.16.2             r1          10.10.11.2       15
10.10.16.1             r1          10.10.11.2       13
10.10.13.1             r1          10.10.11.2        2
10.10.15.1             r1          10.10.11.2        8
10.10.15.2             r1          10.10.11.2       13
mininext>
```

R1

```
2018-11-15 09:45:59.793
Destination Network    Next Hop    Next Hop IP      Cost
--------------------  ----------  ------------  ------
10.10.11.1             Direct      -                 2
10.10.12.2             Direct      -                10
10.10.13.2             Direct      -                 6
10.10.14.1             r2          10.10.12.2       10
10.10.14.2             r3          10.10.13.2       11
10.10.16.2             r3          10.10.13.2       13
10.10.16.1             r3          10.10.13.2       11
10.10.15.1             r3          10.10.13.2        6
10.10.15.2             r3          10.10.13.2       11
mininext>
```

R2

```
2018-11-15 09:46:45.348
Destination Network    Next Hop    Next Hop IP      Cost
---------------------  ----------  -------------  ------
10.10.14.2             Direct      -                   4
10.10.12.1             Direct      -                  10
10.10.11.1             r1          10.10.12.1         12
10.10.11.2             r1          10.10.12.1         10
10.10.13.2             r4          10.10.14.2          9
10.10.16.2             r4          10.10.14.2          6
10.10.16.1             r4          10.10.14.2          4
10.10.13.1             r1          10.10.12.1         10
10.10.15.1             r4          10.10.14.2          9
10.10.15.2             r4          10.10.14.2          4
mininext>
```

R3

```
2018-11-15 09:47:21.488
Destination Network    Next Hop    Next Hop IP      Cost
---------------------  ----------  -------------  ------
10.10.15.2             Direct      -                   5
10.10.13.1             Direct      -                   6
10.10.14.1             r4          10.10.15.2          9
10.10.11.1             r1          10.10.13.1          8
10.10.11.2             r1          10.10.13.1          6
10.10.14.2             r4          10.10.15.2          5
10.10.12.2             r4          10.10.15.2          9
10.10.12.1             r1          10.10.13.1          6
10.10.16.2             r4          10.10.15.2          7
10.10.16.1             r4          10.10.15.2          5
mininext>
```

R4

```
2018-11-15 09:47:42.360
Destination Network    Next Hop    Next Hop IP      Cost
---------------------  ----------  -------------  ------
10.10.16.2             Direct      -                   2
10.10.14.1             Direct      -                   4
10.10.15.1             Direct      -                   5
10.10.11.1             r3          10.10.15.1         13
10.10.11.2             r3          10.10.15.1         11
10.10.12.2             r2          10.10.14.1          4
10.10.12.1             r3          10.10.15.1         11
10.10.13.2             r3          10.10.15.1          5
10.10.13.1             r3          10.10.15.1         11
mininext>
```

H2

```
2018-11-15 09:48:05.782
Destination Network    Next Hop    Next Hop IP      Cost
--------------------   ----------  ------------   ------
10.10.16.1             Direct      -                   2
10.10.14.1             r4          10.10.16.1          6
10.10.11.1             r4          10.10.16.1         15
10.10.11.2             r4          10.10.16.1         13
10.10.14.2             r4          10.10.16.1          2
10.10.12.2             r4          10.10.16.1          6
10.10.12.1             r4          10.10.16.1         13
10.10.13.2             r4          10.10.16.1          7
10.10.13.1             r4          10.10.16.1         13
10.10.15.1             r4          10.10.16.1          7
10.10.15.2             r4          10.10.16.1          2
mininext>
```

Part C.2. Change of weight between r1-r3

a) Time taken to converge:

For estimating convergence when cost changes, we modify the cost files of the two (and take the time after the second modification). We then look at the routing table of all nodes and see which was updated last the new cost. In this case it is **h2.**

Time taken to converge: This calculation is a little problematic. The output from *python* is in UTC, while the output from the *date* command is in PST (UTC-8). After calculating it, the difference is in the order of : **~1s**. But this is random and depends on the polling frequency in the code, currently set to 3s.

```
root@mininet-vm:~/hw/miniNExT/examples/quagga-ixp/configs/r3# vi bf.json
root@mininet-vm:~/hw/miniNExT/examples/quagga-ixp/configs/r3# date +%s.%N
1542262319.429086973
```

```
2018-11-15 06:11:59.329
Destination Network    Next Hop    Next Hop IP      Cost
---------------------  ----------  -------------  ------
10.10.16.1             Direct      -                   2
10.10.14.1             r4          10.10.16.1          6
10.10.11.1             r4          10.10.16.1         15
10.10.11.2             r4          10.10.16.1         13
10.10.14.2             r4          10.10.16.1          2
10.10.12.2             r4          10.10.16.1          6
10.10.12.1             r4          10.10.16.1         13
10.10.13.2             r4          10.10.16.1          7
10.10.13.1             r4          10.10.16.1          8
10.10.15.1             r4          10.10.16.1          7
10.10.15.2             r4          10.10.16.1          2
('Sending data to: ', u'r4', '{"10.10.14.1": ["r4", 6], "10.10.11.1
0.10.13.2": ["r4", 7], "10.10.16.2": ["h2", 0], "10.10.16.1": ["r4"
('Received: ', '{"10.10.14.1": ["r2", 4], "10.10.11.1": ["r3", 8],
, 5], "10.10.16.2": ["h2", 2], "10.10.16.1": ["r4", 0], "10.10.13.1
{u'r4': u'10.10.16.1'}
The DV was changed. So we send the updated costs to everyone
2018-11-15 06:11:59.339
Destination Network    Next Hop    Next Hop IP      Cost
---------------------  ----------  -------------  ------
10.10.16.1             Direct      -                   2
10.10.14.1             r4          10.10.16.1          6
10.10.11.1             r4          10.10.16.1         10
10.10.11.2             r4          10.10.16.1          8
10.10.14.2             r4          10.10.16.1          2
10.10.12.2             r4          10.10.16.1          6
10.10.12.1             r4          10.10.16.1          8
10.10.13.2             r4          10.10.16.1          7
10.10.13.1             r4          10.10.16.1          8
10.10.15.1             r4          10.10.16.1          7
10.10.15.2             r4          10.10.16.1          2
```

b) Routing table at each node:

h1

```
2018-11-15 09:52:12.841
Destination Network    Next Hop    Next Hop IP      Cost
---------------------  ----------  -------------  ------
10.10.11.2             Direct      -                   2
10.10.14.1             r1          10.10.11.2         12
10.10.14.2             r1          10.10.11.2          8
10.10.12.2             r1          10.10.11.2         12
10.10.12.1             r1          10.10.11.2          2
10.10.13.2             r1          10.10.11.2          3
10.10.16.2             r1          10.10.11.2         10
10.10.16.1             r1          10.10.11.2          8
10.10.13.1             r1          10.10.11.2          2
10.10.15.1             r1          10.10.11.2          3
10.10.15.2             r1          10.10.11.2          8
mininext>
```

R1

```
2018-11-15 09:55:40.772
Destination Network    Next Hop    Next Hop IP      Cost
---------------------  ----------  -------------  ------
10.10.11.1             Direct      -                   2
10.10.12.2             Direct      -                  10
10.10.13.2             Direct      -                   1
10.10.14.1             r2          10.10.12.2         10
10.10.14.2             r3          10.10.13.2          6
10.10.16.2             r3          10.10.13.2          8
10.10.16.1             r3          10.10.13.2          6
10.10.15.1             r3          10.10.13.2          1
10.10.15.2             r3          10.10.13.2          6
mininext>
```

R2

```
2018-11-15 09:56:01.862
Destination Network    Next Hop    Next Hop IP      Cost
---------------------  ----------  -------------  ------
10.10.14.2             Direct      -                   4
10.10.12.1             Direct      -                  10
10.10.11.1             r1          10.10.12.1         12
10.10.11.2             r1          10.10.12.1         10
10.10.13.2             r4          10.10.14.2          9
10.10.16.2             r4          10.10.14.2          6
10.10.16.1             r4          10.10.14.2          4
10.10.13.1             r1          10.10.12.1         10
10.10.15.1             r4          10.10.14.2          9
10.10.15.2             r4          10.10.14.2          4
mininext>
```

R3

```
2018-11-15 09:56:31.702
Destination Network    Next Hop    Next Hop IP      Cost
---------------------  ----------  -------------  ------
10.10.15.2             Direct      -                   5
10.10.13.1             Direct      -                   1
10.10.14.1             r4          10.10.15.2          9
10.10.11.1             r1          10.10.13.1          3
10.10.11.2             r1          10.10.13.1          1
10.10.14.2             r4          10.10.15.2          5
10.10.12.2             r4          10.10.15.2          9
10.10.12.1             r1          10.10.13.1          1
10.10.16.2             r4          10.10.15.2          7
10.10.16.1             r4          10.10.15.2          5
mininext>
```

R4

```
2018-11-15 09:56:53.180
Destination Network    Next Hop    Next Hop IP      Cost
--------------------  ----------  -------------  ------
10.10.16.2            Direct      -                   2
10.10.14.1            Direct      -                   4
10.10.15.1            Direct      -                   5
10.10.11.1            r3          10.10.15.1          8
10.10.11.2            r3          10.10.15.1          6
10.10.12.2            r2          10.10.14.1          4
10.10.12.1            r3          10.10.15.1          6
10.10.13.2            r3          10.10.15.1          5
10.10.13.1            r3          10.10.15.1          6
mininext>
```

H2

```
2018-11-15 09:57:17.282
Destination Network    Next Hop    Next Hop IP      Cost
--------------------  ----------  -------------  ------
10.10.16.1            Direct      -                   2
10.10.14.1            r4          10.10.16.1          6
10.10.11.1            r4          10.10.16.1         10
10.10.11.2            r4          10.10.16.1          8
10.10.14.2            r4          10.10.16.1          2
10.10.12.2            r4          10.10.16.1          6
10.10.12.1            r4          10.10.16.1          8
10.10.13.2            r4          10.10.16.1          7
10.10.13.1            r4          10.10.16.1          8
10.10.15.1            r4          10.10.16.1          7
10.10.15.2            r4          10.10.16.1          2
mininext>
```

Part C.3. Negative Weights

Since we have exactly one link with a negative weight, the Bellman-Ford algorithm can handle it. But what does a negative weight mean? There could be two possible scenarios:

1) Link gone bad: A rouge/incorrect link could possible send out negative costs. In this scenario, the best thing to do would be to ignore the link and pretend it did not exist.
2) Deliberately configured so that the path is preferred over others: In this scenario, it could mean that the path with a negative is a “high priority edge”. Then as long as there are no negative cycles, it is okay to have such links.

In conclusion, unless configured to accept negative paths, in the general scenario it is best to ignore a negative cost link.

References:

1. https://www.nongnu.org/quagga/docs/docs-multi/Starting-and-Stopping-ripd.html#Starting-and-Stopping-ripd
2. https://www.linux.com/learn/intro-to-linux/2018/3/dynamic-linux-routing-quagga
3. https://www.brianlinkletter.com/how-to-build-a-network-of-linux-routers-using-quagga/
4. https://cyruslab.net/2012/05/12/install-and-start-quagga/
5. https://wiki.gentoo.org/wiki/Quagga#Show_routing_table