## Part C:

a) Routing protocol code (server.py) is attached.
b) Below is the log when the H1 node is active and each time the distance vector table at H1 gets updated, I print the distance vector table as well as the time. Distance vector table is in format :

"Destination" | "Cost" | "Next Hop"

We can see the time taken by RIP-lite at node H1 is around **2.5 seconds** to converge (1524515758.66 - 1524515756.05). The shortest path from H1 to H2 is 15 and the next hop is R1. Below is the screenshot of log.

```
root@mininet-vm:/# python client.py H1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% H1 is active at 1524515756.05 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Connected by ('172.0.1.2', 41032)
```

```
Connected by ('172.0.1.2', 41064)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Update Routing Table at H1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
R4 | 13 | R1
R1 | 2 | R1
R2 | 12 | R1
R3 | 8 | R1
H2 | 15 | R1
H1 | 0 | H1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Updated Routing Table at 1524515758.66 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

I have attached the routing table update logs at each node for your reference.
For calculating time, whenever I routing table gets updated, I print the time with it. Even I print the time when the node/router becomes active. So after some time, there is no update at any of the node, this means the graph has converged. So the difference in them is the total time taken to find the shortest path by RIP-lite.

c)
Application layer routing table is in format (in all the screenshot follow this convention)

**"Destination" | "Cost" | "Next Hop"**

**H1:**

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Update Routing Table at H1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
R4 | 13 | R1
R1 | 2 | R1
R2 | 12 | R1
R3 | 8 | R1
H2 | 15 | R1
H1 | 0 | H1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Updated Routing Table at 1524515758.66 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

**R1:**

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Update Routing Table at R1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
R4 | 11 | R3
R1 | 0 | R1
R2 | 10 | R2
R3 | 6 | R3
H2 | 13 | R3
H1 | 2 | H1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Updated Routing Table at 1524515758.65 %%%%%%%%%%%%%%%%%%%%%%%%%
```

**R2:**

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Updated Routing Table at 1524515758.41 %%%%%%%%%%%%%%%%%%%%%%%%%
Connected by ('175.0.1.2', 47123)
Connected by ('173.0.1.1', 42314)
Connected by ('173.0.1.1', 42319)
Connected by ('173.0.1.1', 42323)
Connected by ('173.0.1.1', 42327)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% R2 is active at 1524515759.36 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

**R3:**

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Updated Routing Table at 1524515758.41 %%%%%%%%%%%%%%%%%%%%%%%%%
Connected by ('176.0.1.2', 41294)
Connected by ('174.0.1.1', 42148)
Connected by ('174.0.1.1', 42153)
Connected by ('174.0.1.1', 42157)
Connected by ('174.0.1.1', 42161)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% R3 is active at 1524515760.45 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

**R4:**

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Update Routing Table at R4 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
R4 | 0 | R4
R1 | 11 | R3
R2 | 4 | R2
R3 | 5 | R3
H2 | 2 | H2
H1 | 13 | R3
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Updated Routing Table at 1524515758.41 %%%%%%%%%%%%%%%%%%%%%%%%%
```

**H2:**

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Update Routing Table at H2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
R4 | 2 | R4
R1 | 13 | R4
R2 | 6 | R4
R3 | 7 | R4
H2 | 0 | H2
H1 | 15 | R4
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Updated Routing Table at 1524515758.43 %%%%%%%%%%%%%%%%%%%%%%%%%
```

C2)

a) Time taken after changing the edge weight of R1-R3 to converge is (1524520689.52-1524520689.46 = 0.6 seconds). I calculated this by comparing the time when the R1 node read its file till when H1 had updated its routing table and no more update is possible in any node. Below are the screenshot for both timestamps at R1 and H1. I have attached all the logs for all nodes before and after update of weight for your reference.

```
Update:

Connected by ('174.0.1.2', 41388)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Update Routing Table at R1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
R4 | 11 | R3
R1 | 0 | R1
R2 | 10 | R2
R3 | 1 | R1
H2 | 13 | R3
H1 | 2 | H1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Updated Routing Table at 1524520689.46 %%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Update Routing Table at H1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
R4 | 8 | R1
R1 | 2 | R1
R2 | 12 | R1
R3 | 3 | R1
H2 | 10 | R1
H1 | 0 | H1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Updated Routing Table at 1524520689.52 %%%%%%%%%%%%%%%%%%%%%%%%
```

**b) Application layer routing protocol:**

**H1:**

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Update Routing Table at H1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
R4 | 8 | R1
R1 | 2 | R1
R2 | 12 | R1
R3 | 3 | R1
H2 | 10 | R1
H1 | 0 | H1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Updated Routing Table at 1524520689.52 %%%%%%%%%%%%%%%%%%%%%%%%
```

**H2:**

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Update Routing Table at H2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
R4 | 2 | R4
R1 | 8 | R4
R2 | 6 | R4
R3 | 7 | R4
H2 | 0 | H2
H1 | 10 | R4
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Updated Routing Table at 1524520689.51 %%%%%%%%%%%%%%%%%%%%%%%%
```

**R1:**

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Update Routing Table at R1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
R4 | 6 | R3
R1 | 0 | R1
R2 | 10 | R2
R3 | 1 | R1
H2 | 8 | R3
H1 | 2 | H1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Updated Routing Table at 1524520689.49 %%%%%%%%%%%%%%%%%%%%%%%%%%
```

**R2:**

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Update Routing Table at R2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
R4 | 4 | R4
R1 | 10 | R1
R2 | 0 | R2
R3 | 9 | R4
H2 | 6 | R4
H1 | 12 | R1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Updated Routing Table at 1524520621.93 %%%%%%%%%%%%%%%%%%%%%%%%%%
```

**R3:**

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Update Routing Table at R3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
R4 | 5 | R4
R1 | 1 | R3
R2 | 9 | R4
R3 | 0 | R3
H2 | 7 | R4
H1 | 3 | R1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Updated Routing Table at 1524520689.48 %%%%%%%%%%%%%%%%%%%%%%%%%%
```

**R4:**

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Update Routing Table at R4 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
R4 | 0 | R4
R1 | 6 | R3
R2 | 4 | R2
R3 | 5 | R3
H2 | 2 | H2
H1 | 8 | R3
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Updated Routing Table at 1524520689.49 %%%%%%%%%%%%%%%%%%%%%%%%%%
```

c) I have taken care of negative edges. Whenever all the cost in the distance vector table are negative, I stop updating the graph i.e stop sending updates to my neighbor. In this way my graph will converge when all my weights are negative. There is no point updating further because it will run infinitely and keep on decreasing weight.

This is the code snippet used at each node/router:

```
if checkIfAllNegativeWeights() == True:
        return
```