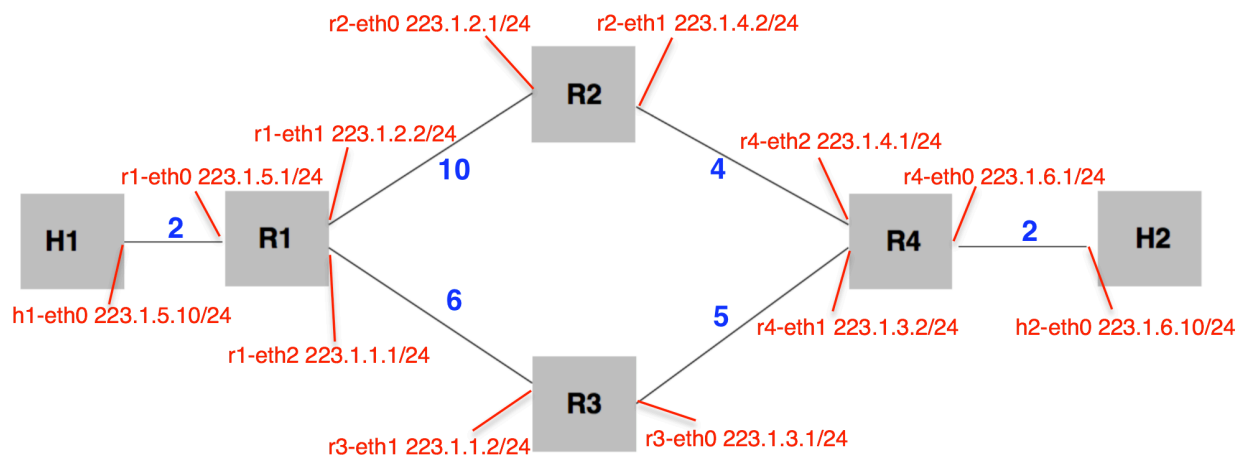


## Table of Contents

C1.....	1
C2.....	11
C3.....	13

### C1.



#### (a) the routing protocol code

---Not in this PDF file---

---See *riplite.py* and *distancevector.py*---

#### (b) the time taken for your protocol to find the shortest path with explanation of how you estimate this

Time = **0.10 second**

Note that I put a 0.1 second of sleeping time while sending distance vectors.

Every host will generate a log. When a host starts initialization, it writes a timestamp into its log. Also, when it updates its distance vector table, it also writes a timestamp into its log. This is a distributed system, where every host runs the application routing protocol asynchronously.

So, the right way to compute the time is to get all the timestamps in all host's log and find the 'first' timestamp and the 'last' timestamp. Then we use the latter one minus the former one.

The 'first' timestamp is 15245512411.68 (appear in h1's log)

The 'last' timestamp is 15245512411.78 (appear in r3's log)

Therefore,  $time = 15245512411.78 - 15245512411.68 = 0.10$

### (c) the application layer routing table at each node

The following routing table comes from *log* file.

#### h1's routing table at each iteration

```

initialization at timestamp = 1524512411.68
Distance(Dest='r1', Cost=2, Next='r1')
Distance(Dest='h2', Cost=9999, Next='')
Distance(Dest='r2', Cost=9999, Next='')
Distance(Dest='r3', Cost=9999, Next='')
Distance(Dest='r4', Cost=9999, Next='')

h1 is listening at port 6666
timestamp = 1524512411.71

h1 received distance vector from r1
update my DV at timestamp = 1524512411.72
Distance(Dest='r1', Cost=2, Next='r1')
Distance(Dest='h2', Cost=9999, Next='')
Distance(Dest='r2', Cost=12, Next='r1')
Distance(Dest='r3', Cost=8, Next='r1')
Distance(Dest='r4', Cost=13, Next='r1')

h1 successfully send dv to neighbors

h1 received distance vector from r1
update my DV at timestamp = 1524512411.73
Distance(Dest='r1', Cost=2, Next='r1')
Distance(Dest='r2', Cost=12, Next='r1')
Distance(Dest='r3', Cost=8, Next='r1')
Distance(Dest='r4', Cost=13, Next='r1')
Distance(Dest='h2', Cost=18, Next='r1')

h1 received distance vector from r1
update my DV at timestamp = 1524512411.74
Distance(Dest='r1', Cost=2, Next='r1')
Distance(Dest='r2', Cost=12, Next='r1')
Distance(Dest='r3', Cost=8, Next='r1')
Distance(Dest='r4', Cost=13, Next='r1')
Distance(Dest='h2', Cost=15, Next='r1')

h1 successfully send dv to neighbors
h1 successfully send dv to neighbors
h1 successfully send dv to neighbors

```

**h2's routing table at each iteration**

initialization at timestamp = 1524512411.7

Distance(Dest='r4', Cost=2, Next='r4')  
Distance(Dest='h1', Cost=9999, Next='')  
Distance(Dest='r1', Cost=9999, Next='')  
Distance(Dest='r2', Cost=9999, Next='')  
Distance(Dest='r3', Cost=9999, Next='')

h2 is listening at port 6666

timestamp = 1524512411.71

h2 received distance vector from r4

update my DV at timestamp = 1524512411.71

Distance(Dest='r4', Cost=2, Next='r4')  
Distance(Dest='h1', Cost=9999, Next='')  
Distance(Dest='r1', Cost=9999, Next='')  
Distance(Dest='r2', Cost=6, Next='r4')  
Distance(Dest='r3', Cost=7, Next='r4')

h2 successfully send dv to neighbors

h2 received distance vector from r4

update my DV at timestamp = 1524512411.74

Distance(Dest='r4', Cost=2, Next='r4')  
Distance(Dest='h1', Cost=9999, Next='')  
Distance(Dest='r2', Cost=6, Next='r4')  
Distance(Dest='r3', Cost=7, Next='r4')  
Distance(Dest='r1', Cost=13, Next='r4')

h2 received distance vector from r4

update my DV at timestamp = 1524512411.75

Distance(Dest='r4', Cost=2, Next='r4')  
Distance(Dest='r2', Cost=6, Next='r4')  
Distance(Dest='r3', Cost=7, Next='r4')  
Distance(Dest='r1', Cost=13, Next='r4')  
Distance(Dest='h1', Cost=18, Next='r4')

h2 successfully send dv to neighbors

h2 received distance vector from r4

update my DV at timestamp = 1524512411.76

Distance(Dest='r4', Cost=2, Next='r4')  
Distance(Dest='r2', Cost=6, Next='r4')  
Distance(Dest='r3', Cost=7, Next='r4')  
Distance(Dest='r1', Cost=13, Next='r4')  
Distance(Dest='h1', Cost=15, Next='r4')

h2 successfully send dv to neighbors

h2 successfully send dv to neighbors

### r1's routing table at each iteration

initialization at timestamp = 1524512411.7

Distance(Dest='h1', Cost=2, Next='h1')

Distance(Dest='r2', Cost=10, Next='r2')

Distance(Dest='r3', Cost=6, Next='r3')

Distance(Dest='h2', Cost=9999, Next='')

Distance(Dest='r4', Cost=9999, Next='')

r1 is listening at port 6666

timestamp = 1524512411.7

r1 received distance vector from h1

update my DV at timestamp = 1524512411.71

Distance(Dest='h1', Cost=2, Next='h1')

Distance(Dest='r2', Cost=10, Next='r2')

Distance(Dest='r3', Cost=6, Next='r3')

Distance(Dest='h2', Cost=9999, Next='')

Distance(Dest='r4', Cost=9999, Next='')

r1 received distance vector from r3

update my DV at timestamp = 1524512411.72

Distance(Dest='h1', Cost=2, Next='h1')

Distance(Dest='r2', Cost=10, Next='r2')

Distance(Dest='r3', Cost=6, Next='r3')

Distance(Dest='h2', Cost=9999, Next='')

Distance(Dest='r4', Cost=11, Next='r3')

r1 received distance vector from r2

update my DV at timestamp = 1524512411.73

Distance(Dest='h1', Cost=2, Next='h1')

Distance(Dest='r2', Cost=10, Next='r2')

Distance(Dest='r3', Cost=6, Next='r3')

Distance(Dest='r4', Cost=11, Next='r3')

Distance(Dest='h2', Cost=16, Next='r2')

r1 received distance vector from h1

update my DV at timestamp = 1524512411.73

Distance(Dest='h1', Cost=2, Next='h1')

Distance(Dest='r2', Cost=10, Next='r2')

Distance(Dest='r3', Cost=6, Next='r3')

Distance(Dest='r4', Cost=11, Next='r3')

Distance(Dest='h2', Cost=16, Next='r2')

r1 received distance vector from h1

update my DV at timestamp = 1524512411.73

```
Distance(Dest='h1', Cost=2, Next='h1')
Distance(Dest='r2', Cost=10, Next='r2')
Distance(Dest='r3', Cost=6, Next='r3')
Distance(Dest='r4', Cost=11, Next='r3')
Distance(Dest='h2', Cost=16, Next='r2')
```

```
r1 received distance vector from r3
update my DV at timestamp = 1524512411.74
Distance(Dest='h1', Cost=2, Next='h1')
Distance(Dest='r2', Cost=10, Next='r2')
Distance(Dest='r3', Cost=6, Next='r3')
Distance(Dest='r4', Cost=11, Next='r3')
Distance(Dest='h2', Cost=13, Next='r3')
```

```
r1 received distance vector from r2
update my DV at timestamp = 1524512411.74
Distance(Dest='h1', Cost=2, Next='h1')
Distance(Dest='r2', Cost=10, Next='r2')
Distance(Dest='r3', Cost=6, Next='r3')
Distance(Dest='r4', Cost=11, Next='r3')
Distance(Dest='h2', Cost=13, Next='r3')
```

```
r1 received distance vector from h1
update my DV at timestamp = 1524512411.75
Distance(Dest='h1', Cost=2, Next='h1')
Distance(Dest='r2', Cost=10, Next='r2')
Distance(Dest='r3', Cost=6, Next='r3')
Distance(Dest='r4', Cost=11, Next='r3')
Distance(Dest='h2', Cost=13, Next='r3')
```

```
r1 received distance vector from r3
update my DV at timestamp = 1524512411.75
Distance(Dest='h1', Cost=2, Next='h1')
Distance(Dest='r2', Cost=10, Next='r2')
Distance(Dest='r3', Cost=6, Next='r3')
Distance(Dest='r4', Cost=11, Next='r3')
Distance(Dest='h2', Cost=13, Next='r3')
```

```
r1 successfully send dv to neighbors
r1 successfully send dv to neighbors
r1 successfully send dv to neighbors
```

### **r2's routing table at each iteration**

```
initialization at timestamp = 1524512411.7
Distance(Dest='r1', Cost=10, Next='r1')
Distance(Dest='r4', Cost=4, Next='r4')
Distance(Dest='h1', Cost=9999, Next='')
Distance(Dest='h2', Cost=9999, Next='')
```

Distance(Dest='r3', Cost=9999, Next='')

r2 is listening at port 6666  
timestamp = 1524512411.7

r2 received distance vector from r4  
update my DV at timestamp = 1524512411.72  
Distance(Dest='r1', Cost=10, Next='r1')  
Distance(Dest='r4', Cost=4, Next='r4')  
Distance(Dest='h1', Cost=9999, Next='')  
Distance(Dest='h2', Cost=6, Next='r4')  
Distance(Dest='r3', Cost=9, Next='r4')

r2 received distance vector from r1  
update my DV at timestamp = 1524512411.74  
Distance(Dest='r1', Cost=10, Next='r1')  
Distance(Dest='r4', Cost=4, Next='r4')  
Distance(Dest='h2', Cost=6, Next='r4')  
Distance(Dest='r3', Cost=9, Next='r4')  
Distance(Dest='h1', Cost=12, Next='r1')

r2 received distance vector from r1  
update my DV at timestamp = 1524512411.75  
Distance(Dest='r1', Cost=10, Next='r1')  
Distance(Dest='r4', Cost=4, Next='r4')  
Distance(Dest='h2', Cost=6, Next='r4')  
Distance(Dest='r3', Cost=9, Next='r4')  
Distance(Dest='h1', Cost=12, Next='r1')

r2 received distance vector from r4  
update my DV at timestamp = 1524512411.75  
Distance(Dest='r1', Cost=10, Next='r1')  
Distance(Dest='r4', Cost=4, Next='r4')  
Distance(Dest='h2', Cost=6, Next='r4')  
Distance(Dest='r3', Cost=9, Next='r4')  
Distance(Dest='h1', Cost=12, Next='r1')

r2 received distance vector from r1  
update my DV at timestamp = 1524512411.75  
Distance(Dest='r1', Cost=10, Next='r1')  
Distance(Dest='r4', Cost=4, Next='r4')  
Distance(Dest='h2', Cost=6, Next='r4')  
Distance(Dest='r3', Cost=9, Next='r4')  
Distance(Dest='h1', Cost=12, Next='r1')

r2 successfully send dv to neighbors

r2 received distance vector from r4  
update my DV at timestamp = 1524512411.76  
Distance(Dest='r1', Cost=10, Next='r1')  
Distance(Dest='r4', Cost=4, Next='r4')  
Distance(Dest='h2', Cost=6, Next='r4')  
Distance(Dest='r3', Cost=9, Next='r4')  
Distance(Dest='h1', Cost=12, Next='r1')

r2 successfully send dv to neighbors

r2 received distance vector from r4  
update my DV at timestamp = 1524512411.77  
Distance(Dest='r1', Cost=10, Next='r1')  
Distance(Dest='r4', Cost=4, Next='r4')  
Distance(Dest='h2', Cost=6, Next='r4')  
Distance(Dest='r3', Cost=9, Next='r4')  
Distance(Dest='h1', Cost=12, Next='r1')

### **r3's routing table at each iteration**

initialization at timestamp = 1524512411.71  
Distance(Dest='r1', Cost=6, Next='r1')  
Distance(Dest='r4', Cost=5, Next='r4')  
Distance(Dest='h1', Cost=9999, Next='')  
Distance(Dest='h2', Cost=9999, Next='')  
Distance(Dest='r2', Cost=9999, Next='')  
  
r3 is listening at port 6666  
timestamp = 1524512411.71  
  
r3 received distance vector from r4  
update my DV at timestamp = 1524512411.73  
Distance(Dest='r1', Cost=6, Next='r1')  
Distance(Dest='r4', Cost=5, Next='r4')  
Distance(Dest='h1', Cost=9999, Next='')  
Distance(Dest='h2', Cost=7, Next='r4')  
Distance(Dest='r2', Cost=9, Next='r4')  
  
r3 successfully send dv to neighbors  
  
r3 received distance vector from r1  
update my DV at timestamp = 1524512411.75  
Distance(Dest='r1', Cost=6, Next='r1')  
Distance(Dest='r4', Cost=5, Next='r4')  
Distance(Dest='h2', Cost=7, Next='r4')  
Distance(Dest='r2', Cost=9, Next='r4')  
Distance(Dest='h1', Cost=8, Next='r1')

r3 received distance vector from r1  
 update my DV at timestamp = 1524512411.76  
 Distance(Dest='r1', Cost=6, Next='r1')  
 Distance(Dest='r4', Cost=5, Next='r4')  
 Distance(Dest='h2', Cost=7, Next='r4')  
 Distance(Dest='r2', Cost=9, Next='r4')  
 Distance(Dest='h1', Cost=8, Next='r1')

r3 successfully send dv to neighbors

r3 received distance vector from r4  
 update my DV at timestamp = 1524512411.76  
 Distance(Dest='r1', Cost=6, Next='r1')  
 Distance(Dest='r4', Cost=5, Next='r4')  
 Distance(Dest='h2', Cost=7, Next='r4')  
 Distance(Dest='r2', Cost=9, Next='r4')  
 Distance(Dest='h1', Cost=8, Next='r1')

r3 received distance vector from r1  
 update my DV at timestamp = 1524512411.76  
 Distance(Dest='r1', Cost=6, Next='r1')  
 Distance(Dest='r4', Cost=5, Next='r4')  
 Distance(Dest='h2', Cost=7, Next='r4')  
 Distance(Dest='r2', Cost=9, Next='r4')  
 Distance(Dest='h1', Cost=8, Next='r1')

r3 successfully send dv to neighbors

r3 received distance vector from r4  
 update my DV at timestamp = 1524512411.77  
 Distance(Dest='r1', Cost=6, Next='r1')  
 Distance(Dest='r4', Cost=5, Next='r4')  
 Distance(Dest='h2', Cost=7, Next='r4')  
 Distance(Dest='r2', Cost=9, Next='r4')  
 Distance(Dest='h1', Cost=8, Next='r1')

r3 received distance vector from r4  
 update my DV at timestamp = 1524512411.78  
 Distance(Dest='r1', Cost=6, Next='r1')  
 Distance(Dest='r4', Cost=5, Next='r4')  
 Distance(Dest='h2', Cost=7, Next='r4')  
 Distance(Dest='r2', Cost=9, Next='r4')  
 Distance(Dest='h1', Cost=8, Next='r1')

#### **r4's routing table at each iteration**

initialization at timestamp = 1524512411.7



```
Distance(Dest='h2', Cost=2, Next='h2')
Distance(Dest='r2', Cost=4, Next='r2')
Distance(Dest='r3', Cost=5, Next='r3')
Distance(Dest='h1', Cost=9999, Next='')
Distance(Dest='r1', Cost=9999, Next='')
```

r4 is listening at port 6666  
timestamp = 1524512411.71

r4 received distance vector from h2  
update my DV at timestamp = 1524512411.71  
Distance(Dest='h2', Cost=2, Next='h2')  
Distance(Dest='r2', Cost=4, Next='r2')  
Distance(Dest='r3', Cost=5, Next='r3')  
Distance(Dest='h1', Cost=9999, Next='')  
Distance(Dest='r1', Cost=9999, Next='')

r4 received distance vector from r3  
update my DV at timestamp = 1524512411.73  
Distance(Dest='h2', Cost=2, Next='h2')  
Distance(Dest='r2', Cost=4, Next='r2')  
Distance(Dest='r3', Cost=5, Next='r3')  
Distance(Dest='h1', Cost=9999, Next='')  
Distance(Dest='r1', Cost=11, Next='r3')

r4 received distance vector from h2  
update my DV at timestamp = 1524512411.74  
Distance(Dest='h2', Cost=2, Next='h2')  
Distance(Dest='r2', Cost=4, Next='r2')  
Distance(Dest='r3', Cost=5, Next='r3')  
Distance(Dest='h1', Cost=9999, Next='')  
Distance(Dest='r1', Cost=11, Next='r3')

r4 successfully send dv to neighbors

r4 received distance vector from r2  
update my DV at timestamp = 1524512411.75  
Distance(Dest='h2', Cost=2, Next='h2')  
Distance(Dest='r2', Cost=4, Next='r2')  
Distance(Dest='r3', Cost=5, Next='r3')  
Distance(Dest='r1', Cost=11, Next='r3')  
Distance(Dest='h1', Cost=16, Next='r2')

r4 received distance vector from h2  
update my DV at timestamp = 1524512411.75  
Distance(Dest='h2', Cost=2, Next='h2')  
Distance(Dest='r2', Cost=4, Next='r2')  
Distance(Dest='r3', Cost=5, Next='r3')

```
Distance(Dest='r1', Cost=11, Next='r3')
Distance(Dest='h1', Cost=16, Next='r2')
```

```
r4 received distance vector from r3
update my DV at timestamp = 1524512411.75
Distance(Dest='h2', Cost=2, Next='h2')
Distance(Dest='r2', Cost=4, Next='r2')
Distance(Dest='r3', Cost=5, Next='r3')
Distance(Dest='r1', Cost=11, Next='r3')
Distance(Dest='h1', Cost=13, Next='r3')
```

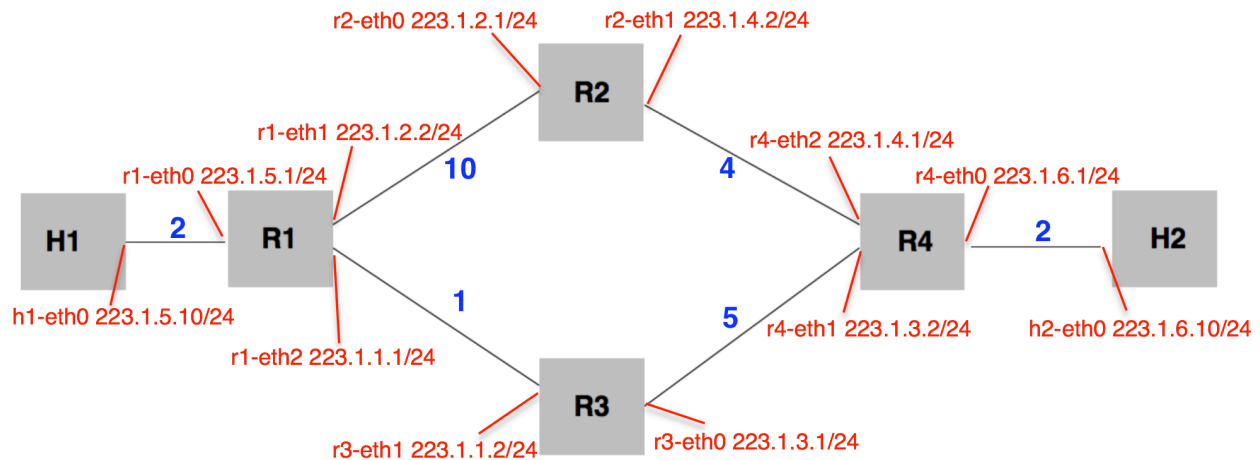
```
r4 received distance vector from r2
update my DV at timestamp = 1524512411.76
Distance(Dest='h2', Cost=2, Next='h2')
Distance(Dest='r2', Cost=4, Next='r2')
Distance(Dest='r3', Cost=5, Next='r3')
Distance(Dest='r1', Cost=11, Next='r3')
Distance(Dest='h1', Cost=13, Next='r3')
```

```
r4 received distance vector from h2
update my DV at timestamp = 1524512411.76
Distance(Dest='h2', Cost=2, Next='h2')
Distance(Dest='r2', Cost=4, Next='r2')
Distance(Dest='r3', Cost=5, Next='r3')
Distance(Dest='r1', Cost=11, Next='r3')
Distance(Dest='h1', Cost=13, Next='r3')
```

```
r4 received distance vector from r3
update my DV at timestamp = 1524512411.76
Distance(Dest='h2', Cost=2, Next='h2')
Distance(Dest='r2', Cost=4, Next='r2')
Distance(Dest='r3', Cost=5, Next='r3')
Distance(Dest='r1', Cost=11, Next='r3')
Distance(Dest='h1', Cost=13, Next='r3')
```

```
r4 successfully send dv to neighbors
r4 successfully send dv to neighbors
r4 successfully send dv to neighbors
```

## C2



Assume that the weight of the link r1-r3 changes from 6 to 1. Estimate the time taken for the protocol to converge.

(a) the time taken for the protocol to converge

Time = **0.06** second

Using the same method as described in C1.

This time is smaller than the time in C1. Since less iterations are needed to converge.

(b) the application layer routing table at each nod

This time only displaying the final distance vector.

**h1's final routing table**

```
update my DV at timestamp = 1524512618.46
Distance(Dest='r1', Cost=2, Next='r1')
Distance(Dest='r2', Cost=12, Next='r1')
Distance(Dest='r3', Cost=3, Next='r1')
Distance(Dest='r4', Cost=8, Next='r1')
Distance(Dest='h2', Cost=10, Next='r1')
```

**h2's final routing table**

```
update my DV at timestamp = 1524512618.47
Distance(Dest='r4', Cost=2, Next='r4')
Distance(Dest='r2', Cost=6, Next='r4')
Distance(Dest='r3', Cost=7, Next='r4')
Distance(Dest='r1', Cost=8, Next='r4')
Distance(Dest='h1', Cost=10, Next='r4')
```

**r1's final routing table**

```
update my DV at timestamp = 1524512618.47
Distance(Dest='h1', Cost=2, Next='h1')
Distance(Dest='r2', Cost=10, Next='r2')
Distance(Dest='r3', Cost=1, Next='r3')
Distance(Dest='r4', Cost=6, Next='r3')
Distance(Dest='h2', Cost=8, Next='r3')
```

**r2's final routing table**

```
update my DV at timestamp = 1524512618.48
Distance(Dest='r1', Cost=10, Next='r1')
Distance(Dest='r4', Cost=4, Next='r4')
Distance(Dest='h2', Cost=6, Next='r4')
Distance(Dest='r3', Cost=9, Next='r4')
Distance(Dest='h1', Cost=12, Next='r1')
```

**r3's final routing table**

```
update my DV at timestamp = 1524512618.47
Distance(Dest='r1', Cost=1, Next='r1')
Distance(Dest='r4', Cost=5, Next='r4')
Distance(Dest='h2', Cost=7, Next='r4')
Distance(Dest='r2', Cost=9, Next='r4')
Distance(Dest='h1', Cost=3, Next='r1')
```

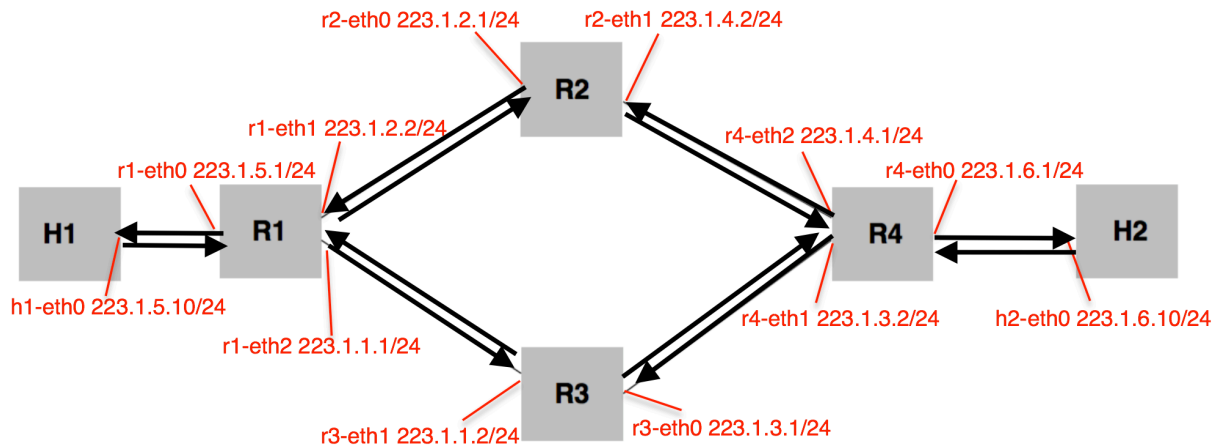
**r4's final routing table**

```
update my DV at timestamp = 1524512618.46
Distance(Dest='h2', Cost=2, Next='h2')
Distance(Dest='r2', Cost=4, Next='r2')
Distance(Dest='r3', Cost=5, Next='r3')
Distance(Dest='r1', Cost=6, Next='r3')
Distance(Dest='h1', Cost=8, Next='r3')
```

## C3

**If one of the links has a negative weight, explain how you handle this situation.**

In directed graph, Bellman-Ford can deal with shortest path problem with negative edges but with no negative cycles. However, the graph in this problem does have cycles. The graph is actually like this, which is bi-directional.



So, if one of the links has a negative weight, there must be a negative cycle. **Therefore, the Bell-Ford algorithm won't work.**

The naïve solution is to add a certain positive value to every edge. So that the negative edge "disappear". However, **this will change the shortest path**. A simple counter example will prove this. The main reason is that a shortest path can have many small edges, while a longer path has a few big edges. So, when adding a certain value, the previous shortest path will have "disadvantage", possibly making it not the shortest path anymore.

So, I think the right solution is **to view this negative link as a link failure**, i.e. the link is down. When implementing, I will make the failure link's weight to infinity.

But this introduces a new problem. Bad news travels slowly. So, this is the count to infinity problem. To solve this problem, we can make infinity not too big. Also, we should apply split horizon-based solution. That is don't send DV update to a neighbor for a destination, where that neighbor is the next hop for that destination. Also, we use poisoned reverse: send such DV updates but with infinite distance metric[1].

Reference:

[1] Prof. Aruna's lecture slide: [lectrouting.pdf](#)