

Please check the above for typos, inconsistencies, etc., there are a lot of details and we could have missed a few.

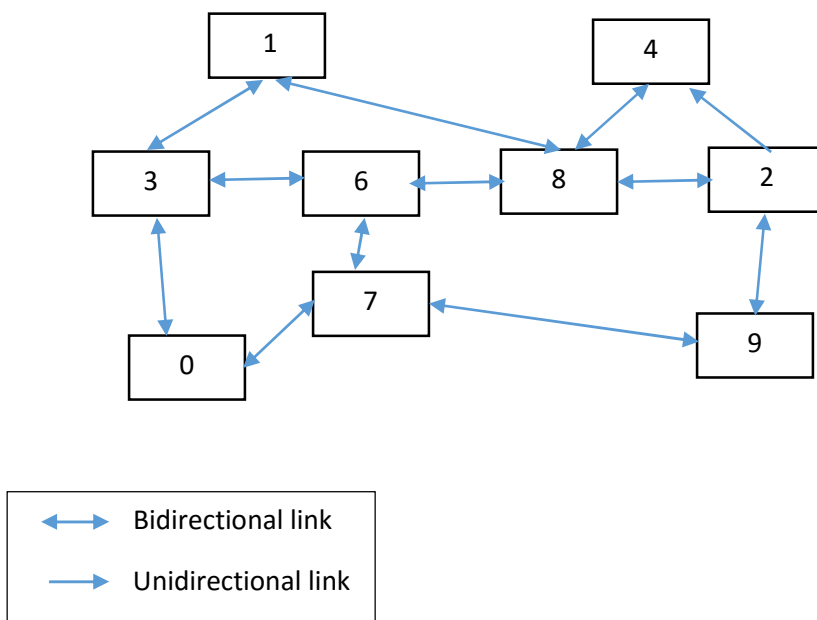
You will receive a maximum of 30 points if your project is unable to do any of the tasks below. This depends on how much effort I think you placed on your code.

If all of the below works, you get a 100

I of course do not guarantee that exactly the same scenarios will be used for grading, but you get the idea.

Scenario 1

Consider the following network:



The topology.txt for this network:

0 UP 0 3

0 UP 3 0

0 UP 0 7

0 UP 7 0

0 UP 1 3

0 UP 3 1

0 UP 1 8

0 UP 8 1

0 UP 2 4

0 UP 2 8

0 UP 8 2

0 UP 2 9

0 UP 9 2

0 UP 3 6

0 UP 6 3

0 UP 4 8

0 UP 8 4

0 UP 6 7

0 UP 7 6

0 UP 6 8

0 UP 8 6

0 UP 7 9

0 UP 9 7

80 DOWN 6 8

80 DOWN 8 6

Assume that we want to send data messages from node 3 to 2 and from node 0 to 1 and from 9 to 2. All other nodes are active nodes but will not be generating any data messages of their own. For this, the shell script will look like this:

```
./node 0 1 "message from 0" 50 &
```

```
./node 1 1 &
```

```
./node 2 2 &
```

```
./node 3 2 "message from 3" 50 &
```

```
./node 4 4 &
```

```
./node 6 6 &
```

```
./node 7 7 &
```

```
./node 8 8 &
```

```
./node 9 2 "message from 9" 25 &
```

```
./controller &
```

We will test the following functionalities:

15 Points: Hello messages are sent between nodes, starting with the unidirectional neighbor list in the initial hellos, until updating hello messages with the bidirectional neighbor list. If a link is unidirectional between neighbors, hello messages should only propagate one way. Hellos should be sent by every node every 5 seconds.

10 Points: Hello messages should have the correct MPR list.

10 Points: After time 80 when the link between 6 and 8 goes down, the hello messages should reflect (after some time) the correct two-hop neighborhood and MPRs.

15 Points: TC message propagation (i.e., check whether it is being broadcast correctly) as well as the contents of the TC message. The TC message should be sent every 10 seconds.

10 Points: Data message propagation in the network. We are looking to make sure route computation is done correctly and the data message travels along the shortest path to the destination.

Scenario 2

10 Points: We use the same topology as before, except that after time 50, the links between 1 and 8 and also between 6 and 8 go down. The data from 3 to 2 is sent at time 90 (later than before) and thus it must follow the new path created after all nodes learn of the topology changes. Thus, the topology.txt file for scenario 2 looks as follows:

0 UP 0 3

0 UP 3 0

0 UP 0 7

0 UP 7 0

0 UP 1 3

0 UP 3 1

0 UP 1 8

0 UP 8 1

0 UP 2 4

0 UP 2 8

0 UP 8 2

0 UP 2 9
0 UP 9 2
0 UP 3 6
0 UP 6 3
0 UP 4 8
0 UP 8 4
0 UP 6 7
0 UP 7 6
0 UP 6 8
0 UP 8 6
0 UP 7 9
0 UP 9 7
50 DOWN 6 8
50 DOWN 8 6
50 DOWN 1 8
50 DOWN 8 1

The shell-script for scenario2 looks as follows:

```
./node 0 1 "message from 0" 50 &  
./node 1 1 &  
./node 2 2 &  
./node 3 2 "message from 3" 90 &  
./node 4 4 &  
./node 6 6 &  
./node 7 7 &  
./node 8 8 &  
./node 9 2 "message from 9" 25 &  
./controller &
```