Computer Networks
CS3205
Assignment 3

OSPF Routing Algorithm

Soumith Basina - CS18B005

Date Submitted: April 20th, 2021

Instructor ..... Prof. Siva Ram Murthy

Contents

1 Objective

The objective of this assignment is to emulate a simplified version of the Open Shortest Path First (OSPF) routing protocol, as explained in the problem statement and observe the results.

2 Introduction

Open Shortest Path First (OSPF) is a routing protocol for Internet Protocol (IP) networks. It uses Link State Routing (LSR) algorithm. The shortest path is calculated based on the cost of the route, making it adaptive to the changing network conditions.

All routers keep sending hello packets, and thus changes in the cost of their links become known to neighboring routers. The information about the cost of a link, is then cascaded through the network because OSPF routers advertise the information they receive from one neighboring router to all other neighboring routers. This process of flooding link state information through the network is known as synchronization Based on this information, all routers continuously update their link state databases with information about the network topology and adjust their routing tables.

The algorithm that is emulated in the assignment follows these principles.

3 Experimental details

The algorithm in this experiment has 3 adjustable parameters.

- hi - Hello interval
- $lsai$ - LSA interval
- $spfi$ - SPF interval

They denote the periods at which HELLO packets, LSA packets, SPF calculation repeat during execution.

3.1 Experimental/Simulation setup

These three parameters can be passed to the `ospf` executable along with the node id, the name of the infile and outfile to generate the outfiles. The infile consists of the information about the network such as number of routers, links and details about links and the range of costs for each link. The outfile consists of routing tables which are printed every $spfi$ seconds.

For the emulation of the network, a program instance should be running for every router to be emulated. That is automated using the bash script `routersetup.sh` to run multiple instances in parallel. The number of routers should be passed as a command-line argument to the script.

3.2 Entities involved and functions in each entity

`ospf.cpp` has the code for the emulation of the algorithm. It takes in parameters through command line and writes the routing table of the given router to an outfile of the specified name.

Sending HELLO, LSA packets and calculating paths should be done periodically and should not interrupt the receiving packets. Therefore, those tasks are run concurrently by spawning a thread for each task. This is done by the function `threadStart()`.

```
void threadStart(function<void()> func, unsigned int interval)
{
    thread([func, interval]() {
        while (true)
        {
            auto x = chrono::steady_clock::now() +
            chrono::milliseconds(interval);
            func();
            this_thread::sleep_until(x);
        }
    }).detach();
}
```

This creates a new thread that executes the given function every `interval` ms. Using this, we create new threads for 3 functions - `sendHello, sendLSA, calculatePaths`.

`routersetup.sh` is a bash script that takes the number of routers as a command-line argument and runs multiple instances of the executable in parallel. All the processes can be killed by using Ctrl + C.

4 Results and Observations

4.1 Example 1

The infile consists of the following

```
8 20
1 0 5 20
1 2 12 30
2 0 13 17
3 1 11 16
3 2 7 19
0 3 2 10
4 5 16 25
4 6 27 31
4 7 21 24
5 6 31 35
5 7 10 15
```

```
6 7 23 27
0 4 1 30
4 1 9 19
1 5 12 14
7 2 13 20
3 6 11 17
4 2 4 7
3 7 10 13
6 2 27 20
```

The routing tables are as the following

```
Routing Table for Node no. 0 at time 10
Destination     Path    Cost
1               01      11
2               02      16
3               03      5
4               04      14
5               015     24
6               036     19
7               037     15
Routing Table for Node no. 0 at time 20
Destination     Path    Cost
1               01      11
2               02      17
3               03      10
4               04      20
5               015     25
6               036     25
7               037     20
```

(a) Node 0

```
Routing Table for Node no. 1 at time 10
Destination     Path    Cost
0               10      11
2               12      13
3               13      13
4               14      15
5               15      12
6               136     27
7               137     23
Routing Table for Node no. 1 at time 20
Destination     Path    Cost
0               10      11
2               12      15
3               13      14
4               14      16
5               15      14
6               136     29
7               137     24
```

(b) Node 1

```
Routing Table for Node no. 2 at time 10
Destination     Path    Cost
0               20      16
1               21      12
3               23      19
4               24      4
5               245     20
6               26      27
7               27      19
Routing Table for Node no. 2 at time 20
Destination     Path    Cost
0               20      17
1               241     15
3               23      14
4               24      5
5               245     21
6               26      29
7               27      19
```

(c) Node 2

```
Routing Table for Node no. 3 at time 10
Destination     Path    Cost
0               30      5
1               31      13
2               32      19
4               304     19
5               375     23
6               36      11
7               37      11
Routing Table for Node no. 3 at time 20
Destination     Path    Cost
0               30      10
1               31      14
2               32      14
4               324     19
5               375     25
6               36      17
7               37      12
```

(d) Node 3

Figure 1: Routing tables for the first topology

Routing Table for Node no. 4 at time 10

Destination	Path	Cost
0	40	14
1	41	9
2	42	4
3	403	19
5	45	19
6	46	30
7	47	21

Routing Table for Node no. 4 at time 20

Destination	Path	Cost
0	40	20
1	41	16
2	42	5
3	423	19
5	45	16
6	46	27
7	47	22

(a) Node 4

Routing Table for Node no. 5 at time 10

Destination	Path	Cost
0	510	23
1	51	12
2	542	23
3	573	24
4	54	19
6	56	33
7	57	14

Routing Table for Node no. 5 at time 20

Destination	Path	Cost
0	510	25
1	51	14
2	542	21
3	573	21
4	54	16
6	56	33
7	57	11

(b) Node 5

Routing Table for Node no. 6 at time 10

Destination	Path	Cost
0	630	16
1	631	24
2	62	27
3	63	11
4	64	30
5	65	33
7	637	22

Routing Table for Node no. 6 at time 20

Destination	Path	Cost
0	630	27
1	631	31
2	62	29
3	63	17
4	64	27
5	65	33
7	67	26

(c) Node 6

Routing Table for Node no. 7 at time 10

Destination	Path	Cost
0	730	16
1	731	24
2	72	19
3	73	11
4	74	21
5	75	14
6	736	22

Routing Table for Node no. 7 at time 20

Destination	Path	Cost
0	730	22
1	751	25
2	72	19
3	73	12
4	74	22
5	75	11
6	76	26

(d) Node 7

Figure 2: Routing tables for the first topology (continued...)

4.2 Example 2

The infile consists of the following

```
8 22
0 1 4 10
1 2 3 9
2 0 6 10
3 1 4 10
3 2 3 9
0 3 6 10
0 4 2 5
4 1 7 20
2 4 3 7
4 3 9 17
0 5 10 15
5 1 13 20
2 5 20 27
5 3 25 26
0 6 12 16
6 1 13 17
2 6 4 6
6 3 1 5
0 7 9 15
7 1 15 20
2 7 19 24
7 3 30 35
```

The routing tables are as follows

```
Routing Table for Node no. 0 at time 10
Destination     Path    Cost
1               01      4
2               02      9
3               03      9
4               04      3
5               05      15
6               036     14
7               07      15
Routing Table for Node no. 0 at time 20
Destination     Path    Cost
1               01      6
2               042     8
3               03      9
4               04      5
5               05      15
6               036     11
7               07      12
```

(a) Node 0

```
Routing Table for Node no. 1 at time 10
Destination     Path    Cost
0               10      4
2               12      3
3               13      4
4               104     7
5               15      13
6               126     8
7               17      15
Routing Table for Node no. 1 at time 20
Destination     Path    Cost
0               10      6
2               12      3
3               13      4
4               124     6
5               15      19
6               136     6
7               107     18
```

(b) Node 1

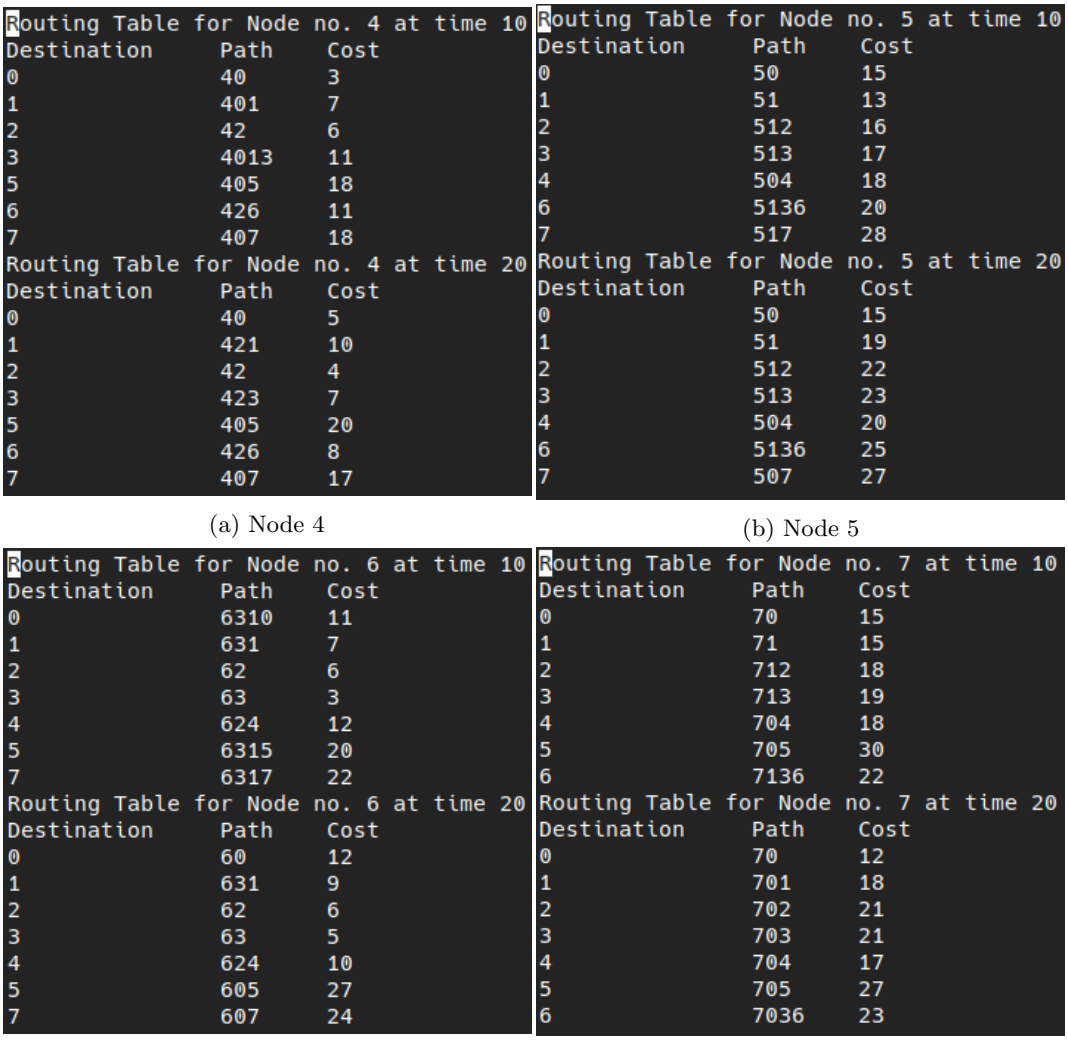
```
Routing Table for Node no. 2 at time 10
Destination     Path    Cost
0               210     7
1               21      3
3               213     7
4               24      6
5               215     16
6               26      5
7               217     18
Routing Table for Node no. 2 at time 20
Destination     Path    Cost
0               20      9
1               21      3
3               23      6
4               24      4
5               215     22
6               26      6
7               207     21
```

(c) Node 2

```
Routing Table for Node no. 3 at time 10
Destination     Path    Cost
0               310     8
1               31      4
2               362     7
4               34      9
5               315     17
6               36      2
7               317     19
Routing Table for Node no. 3 at time 20
Destination     Path    Cost
0               30      9
1               31      7
2               32      6
4               324     9
5               315     23
6               36      5
7               307     21
```

(d) Node 3

Figure 3: Routing tables for the second topology



(a) Node 4

(b) Node 5

(c) Node 6

(d) Node 7

Figure 4: Routing tables for the second topology (continued...)

4.3 Observations

As we can observe, the paths and costs have been changing as time passes. This is one of the advantages that OSPF has. It can adjust the routing tables to accommodate for the changing network conditions. And the process is quick too as the LSA packets flood the network very quickly with the cost of some network overhead.

5 Learnings

This assignment has been helpful in

- Learning some insights about the OSPF protocol and how routing in general works.
- Learning Bash scripting and basic threading in C++.

6 Conclusion

OSPF is a widely deployed routing protocol that can converge a network in a few seconds and guarantee loop-free paths. Having a quick routing protocol that is adaptive to the network conditions is vital to maintain efficient communication within the network. With this assignment, we have gained insights into how the algorithm works by emulating how the communications occur between the routers.

7 References

- https://en.wikipedia.org/wiki/Open_Shortest_Path_First