

Assignment 3

Aim : To implement a simple version of the Open Shortest Path First(OSPF) routing protocol.

Introduction :

Internet topology roughly organised as a two level hierarchy

- 1) First lower level - autonomous systems (AS's)
- 2) Second level - inter-connected AS's

Each Autonomous system runs an intra-domain routing protocol. It is based on unreliable datagram delivery, in particular now we are dealing with Link state routing(OSPF).In this each node knows it's connectivity and cost to direct neighbour, every node tells this information to all other nodes by flooding. In the end every node learns the topology of the network and finds the shortest path to every router.

Experimental details:

- Experimental/Simulation setup :
 1. Here we are running a count of number of node processes at same time
 2. With in each process we are keeping 4 threads one each for spf,hello,lsa,receive functions
 3. On the same machine the processes are communicating through UDP.
- Entities involved and functions in each entity:
 - There are three important functions send_lsa , send_hello , dijkstras, these are executed parallely by different threads.
 - Send_lsa send hello packets to neighbours, send_lsa will send lsa packets to its neighbours and dijkstra's compute the shortest path based on the knowledge it has , these three functions are executed parallely by threads.
 - A bash script which invokes different processes and makes them execute and generates output files which are shortest path tables

Results and Observations:

Here I am pasting the screenshots of tables that are generated in two sessions for various inputs, here I am pasting only 2 tables per output file of node , please see Session_1 files,Session_2 files directory for complete tables(10).

Here the hyperparameters Hello interval , LSA interval , SPF interval are default values 1,5,20.

Input 1:

8 22

0 1 4 1

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

Routing table for Node no:0 at Time20		
-Destination	-PATH-	-Cost-
1	0-1	7
2	0-2	8
3	0-3	7
4	0-4	4
5	0-5	11
6	0-3-6	10
7	0-7	10
Routing table for Node no:0 at Time40		
-Destination	-PATH-	-Cost-
1	0-1	10
2	0-4-2	5
3	0-3	6
4	0-4	2
5	0-5	12
6	0-4-2-6	9
7	0-7	11

Routing table for Node no:1 at Time20		

-Destination	-PATH-	-Cost-
0	1-0	--9--
2	1-2	--6--
3	1-3	--6--
4	1-2-4	--11--
5	1-5	--15--
6	1-3-6	--9--
7	1-7	--15--

Routing table for Node no:1 at Time40		

-Destination	-PATH-	-Cost-
0	1-0	--7--
2	1-2	--9--
3	1-3	--6--
4	1-4	--9--
5	1-5	--19--
6	1-3-6	--11--
7	1-0-7	--19--

Routing table for Node no:2 at Time20		

-Destination	-PATH-	-Cost-
0	2-4-0	--7--
1	2-1	--6--
3	2-3	--7--
4	2-4	--5--
5	2-1-5	--21--
6	2-6	--5--
7	2-4-0-7	--16--

Routing table for Node no:2 at Time40		

-Destination	-PATH-	-Cost-
0	2-4-0	--5--
1	2-1	--9--
3	2-3	--7--
4	2-4	--3--
5	2-4-0-5	--20--
6	2-6	--4--
7	2-4-0-7	--17--

Routing table for Node no:3 at Time20		
-Destination-	-PATH-	-Cost-
0	3-0	8
1	3-1	6
2	3-2	7
4	3-0-4	10
5	3-1-5	21
6	3-6	3
7	3-0-7	17
Routing table for Node no:3 at Time40		
-Destination-	-PATH-	-Cost-
0	3-0	6
1	3-1	6
2	3-2	7
4	3-0-4	8
5	3-0-5	21
6	3-6	5
7	3-0-7	18

Routing table for Node no:4 at Time20		
-Destination-	-PATH-	-Cost-
0	4-0	4
1	4-0-1	11
2	4-2	5
3	4-0-3	11
5	4-0-5	15
6	4-2-6	10
7	4-0-7	14
Routing table for Node no:4 at Time40		
-Destination-	-PATH-	-Cost-
0	4-0	2
1	4-1	9
2	4-2	3
3	4-0-3	8
5	4-0-5	17
6	4-2-6	7
7	4-0-7	14

Routing table for Node no:5 at Time20		
-Destination-	-PATH-	-Cost-
0	5-0	--11--
1	5-1	--15--
2	5-0-2	--19--
3	5-0-3	--18--
4	5-0-4	--15--
6	5-0-3-6	--21--
7	5-0-7	--21--

Routing table for Node no:5 at Time40		
-Destination-	-PATH-	-Cost-
0	5-0	--12--
1	5-1	--15--
2	5-0-4-2	--17--
3	5-0-3	--18--
4	5-0-4	--14--
6	5-0-4-2-6	--21--
7	5-0-7	--23--

Routing table for Node no:6 at Time20		
-Destination-	-PATH-	-Cost-
0	6-3-0	--10--
1	6-3-1	--9--
2	6-2	--5--
3	6-3	--3--
4	6-2-4	--10--
5	6-3-0-5	--21--
7	6-3-0-7	--20--

Routing table for Node no:6 at Time40		
-Destination-	-PATH-	-Cost-
0	6-2-4-0	--9--
1	6-3-1	--11--
2	6-2	--4--
3	6-3	--5--
4	6-2-4	--7--
5	6-2-4-0-5	--21--
7	6-2-4-0-7	--20--

Routing table for Node no:7 at Time20		
-Destination-	-PATH-	-Cost-
0	7-0	10
1	7-1	15
2	7-0-2	18
3	7-0-3	17
4	7-0-4	14
5	7-0-5	21
6	7-0-3-6	20
Routing table for Node no:7 at Time40		
-Destination-	-PATH-	-Cost-
0	7-0	12
1	7-0-1	19
2	7-0-4-2	17
3	7-0-3	18
4	7-0-4	14
5	7-0-5	27
6	7-0-4-2-6	21

Input 2:

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 20 2

Routing table for Node no:0 at Time20		
-Destination	-PATH-	-Cost-
1	0-1	14
2	0-4-2	14
3	0-3	6
4	0-4	8
5	0-1-5	27
6	0-3-6	22
7	0-3-7	17
Routing table for Node no:0 at Time40		
-Destination	-PATH-	-Cost-
1	0-1	16
2	0-2	15
3	0-3	5
4	0-4	17
5	0-1-5	28
6	0-3-6	18
7	0-3-7	18

Routing table for Node no:1 at Time20		
-Destination-	-PATH-	-Cost-
0	1-0	8
2	1-2	21
3	1-3	11
4	1-0-4	16
5	1-5	13
6	1-3-6	27
7	1-3-7	22
Routing table for Node no:1 at Time40		
-Destination-	-PATH-	-Cost-
0	1-0	16
2	1-2	21
3	1-3	14
4	1-4	17
5	1-5	12
6	1-3-6	27
7	1-5-7	24

Routing table for Node no:2 at Time20		
-Destination-	-PATH-	-Cost-
0	2-4-0	14
1	2-1	21
3	2-3	14
4	2-4	6
5	2-7-5	29
6	2-6	26
7	2-7	19
Routing table for Node no:2 at Time40		
-Destination-	-PATH-	-Cost-
0	2-0	15
1	2-1	21
3	2-3	11
4	2-4	6
5	2-7-5	26
6	2-6	24
7	2-7	14

Routing table for Node no:3 at Time20		

-Destination	-----PATH-----	-Cost-
0	3-0	6
1	3-1	11
2	3-2	14
4	3-0-4	14
5	3-7-5	21
6	3-6	16
7	3-7	11

Routing table for Node no:3 at Time40		

-Destination	-----PATH-----	-Cost-
0	3-0	5
1	3-1	11
2	3-2	11
4	3-2-4	17
5	3-1-5	23
6	3-6	13
7	3-7	13

Routing table for Node no:4 at Time20		

-Destination	-----PATH-----	-Cost-
0	4-0	8
1	4-0-1	16
2	4-2	6
3	4-0-3	14
5	4-5	25
6	4-6	28
7	4-7	24

Routing table for Node no:4 at Time40		

-Destination	-----PATH-----	-Cost-
0	4-0	17
1	4-1	17
2	4-2	6
3	4-2-3	17
5	4-5	23
6	4-6	30
7	4-2-7	20

Routing table for Node no:5 at Time20		

-Destination	-PATH-	-Cost-
0	5-1-0	21
1	5-1	13
2	5-4-2	31
3	5-1-3	24
4	5-4	25
6	5-6	32
7	5-7	15

Routing table for Node no:5 at Time40		

-Destination	-PATH-	-Cost-
0	5-1-0	28
1	5-1	12
2	5-7-2	26
3	5-1-3	23
4	5-4	23
6	5-6	35
7	5-7	12

Routing table for Node no:6 at Time20		

-Destination	-PATH-	-Cost-
0	6-3-0	22
1	6-3-1	27
2	6-2	26
3	6-3	16
4	6-4	28
5	6-5	32
7	6-7	23

Routing table for Node no:6 at Time40		

-Destination	-PATH-	-Cost-
0	6-3-0	18
1	6-3-1	24
2	6-2	24
3	6-3	13
4	6-4	30
5	6-5	35
7	6-7	23

Routing table for Node no:7 at Time20		
-Destination-	-PATH-	-Cost-
0	7-3-0	19
1	7-5-1	23
2	7-2	19
3	7-3	13
4	7-4	24
5	7-5	10
6	7-6	23
Routing table for Node no:7 at Time40		
-Destination-	-PATH-	-Cost-
0	7-3-0	18
1	7-5-1	24
2	7-2	14
3	7-3	13
4	7-2-4	20
5	7-5	12
6	7-6	23

Observations:

The values are changing with time , and shortest paths computed at same time by different nodes are not changing much because we are executing them parallelly, In some cases they are changing because the link state advertisement at one node may not be reached the other node before the spf interval starts.

Additional Details:

To check the correctness of the shortest path code I used test cases where low cost is equal to high cost and constructed multiple cases and I am successful regarding correctness.

I have used mutex_locks and done synchronisation when two threads are trying to access the same memory variable.