Computer Networks CA#3

Aryan Soltani Mohammad 810198558
Moein Karami 810198540

In this project two famous algorithms for routing algorithms had been implemented and time difference had been computed for finding the decent algorithm.

Output of Show Command

```
show
u|v | 1 2 3 4 5 6 7 8 9 10 11 12 13
-----------------------------------------------------
1 | 0 -1 -1 -1 6 -1 7 -1 -1 -1 -1 -1 -1
2 | -1 0 -1 -1 -1 2 13 -1 -1 -1 -1 -1 -1
3 | -1 -1 0 -1 -1 -1 1 4 -1 -1 -1 8 -1
4 | -1 -1 -1 0 -1 19 -1 -1 -1 11 -1 -1 -1
5 | 6 -1 -1 -1 0 -1 -1 -1 -1 3 -1 -1 -1
6 | -1 2 -1 19 -1 0 -1 17 -1 -1 25 -1 4
7 | 7 13 1 -1 -1 -1 0 -1 -1 -1 -1 8 -1
8 | -1 -1 4 -1 -1 17 -1 0 -1 -1 16 -1 -1
9 | -1 -1 -1 -1 -1 -1 -1 -1 0 -1 -1 5 7
10 | -1 -1 -1 11 3 -1 -1 -1 -1 0 -1 12 -1
11 | -1 -1 -1 -1 -1 25 -1 16 -1 -1 0 -1 -1
12 | -1 -1 8 -1 -1 -1 8 -1 5 12 -1 0 -1
13 | -1 -1 -1 -1 -1 4 -1 -1 7 -1 -1 -1 0
```

Topology Command: topology 13-9-7 13-6-4 9-12-5 12-7-8 7-2-13 2-6-2 6-8-17 6-11-25 8-11-16 6-4-19 10-4-11 10-5-3 5-1-6 1-7-7 12-3-8 8-3-4 10-12-12 3-7-1

show

u|v | 1 2 3 4 5 6 7 8 9 10 11 12 13
---

1 | 0 -1 -1 -1 6 -1 7 -1 -1 -1 -1 -1 -1
2 | -1 0 -1 -1 -1 2 13 -1 -1 -1 -1 -1 -1
3 | -1 -1 0 -1 -1 -1 1 4 -1 -1 -1 8 -1
4 | -1 -1 -1 0 -1 19 -1 -1 -1 11 -1 -1 -1
5 | 6 -1 -1 -1 0 -1 -1 -1 -1 3 -1 -1 -1
6 | -1 2 -1 19 -1 0 -1 17 -1 -1 25 -1 4
7 | 7 13 1 -1 -1 -1 0 -1 -1 -1 -1 8 -1
8 | -1 -1 4 -1 -1 17 -1 0 -1 -1 16 -1 -1
9 | -1 -1 -1 -1 -1 -1 -1 -1 0 -1 -1 5 7
10 | -1 -1 -1 11 3 -1 -1 -1 -1 0 -1 12 -1
11 | -1 -1 -1 -1 -1 25 -1 16 -1 -1 0 -1 -1
12 | -1 -1 8 -1 -1 -1 8 -1 5 12 -1 0 -1
13 | -1 -1 -1 -1 -1 4 -1 -1 7 -1 -1 -1 0

LSRP

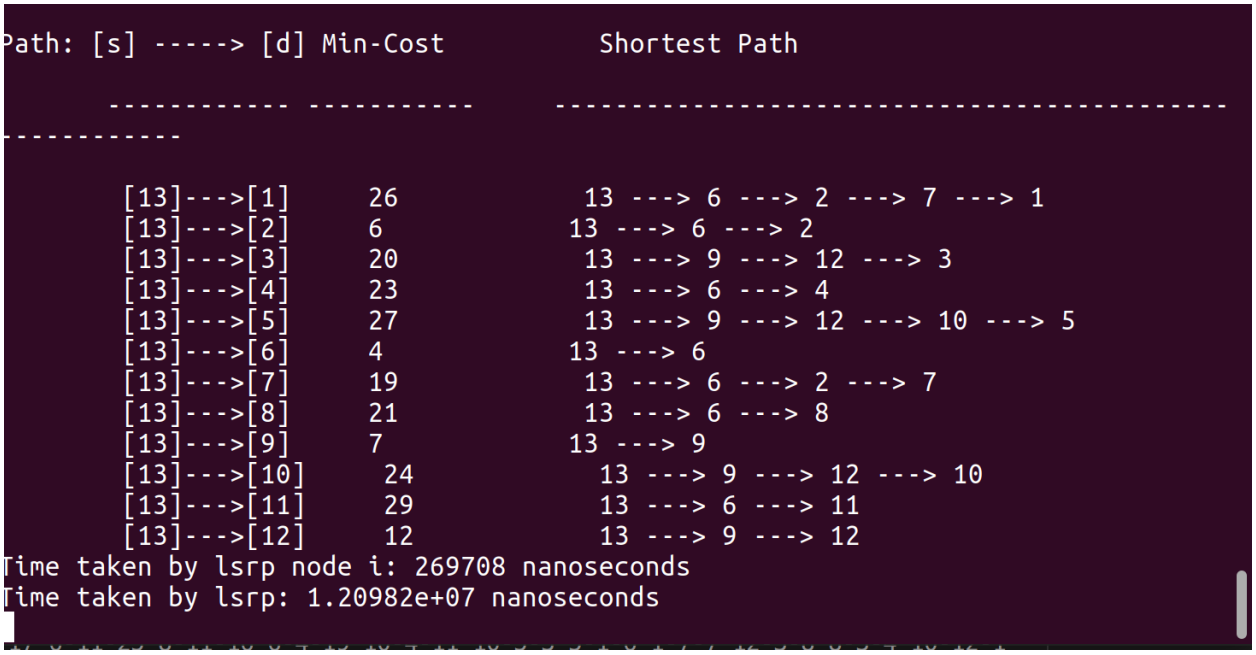
The first Algorithm is LSRP(Link State Algorithm) which executes a dijkstra algorithm in the given source node for finding the shortest path from source to each of the nodes in the topology.

Running LSRP for src = 1

```
lsrp 1
        Iter 1:

Dest    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |

Cost    | 0 | -1 | -1 | -1 | 6 | -1 | 7 | -1 | -1 | -1 | -1 | -1 | -1 |

-----------------------------------------------------------------

        Iter 2:

Dest    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |

Cost    | 0 | -1 | -1 | -1 | 6 | -1 | 7 | -1 | -1 | 9 | -1 | -1 | -1 |

-----------------------------------------------------------------

        Iter 3:

Dest    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
```

```
Cost    | 0 | 20 | 8 | 20 | 6 | 22 | 7 | 12 | 20 | 9 | 28 | 15 | 26 |

-----------------------------------------------------------------

Path: [s] -----> [d] Min-Cost          Shortest Path

       ----------- -----------     ----------------------------------------
------------

        [1]--->[2]     20            1 ---> 7 ---> 2
        [1]--->[3]     8            1 ---> 7 ---> 3
        [1]--->[4]     20            1 ---> 5 ---> 10 ---> 4
        [1]--->[5]     6            1 ---> 5
        [1]--->[6]     22            1 ---> 7 ---> 2 ---> 6
        [1]--->[7]     7            1 ---> 7
        [1]--->[8]     12            1 ---> 7 ---> 3 ---> 8
        [1]--->[9]     20            1 ---> 7 ---> 12 ---> 9
        [1]--->[10]     9            1 ---> 5 ---> 10
        [1]--->[11]     28            1 ---> 7 ---> 3 ---> 8 ---> 11
        [1]--->[12]     15            1 ---> 7 ---> 12
        [1]--->[13]     26            1 ---> 7 ---> 2 ---> 6 ---> 13
Time taken by lsrp: 760381 nanoseconds
```

Running for all source



DVRP

DVRP(Distance Vector) Algorithm is an algorithm similar to bellman ford and it execute till it converges and in each loop every node give its neighbors its routing table and the node update itself with the given data.

DVRP for one node:

```
Dest Next Hop Dist Shortest Path

------------------------------------------------------------------------
------

1       1         0            [ 1 ]
2       7         20            [ 1 ---> 7 ---> 2 ]
3       7         8            [ 1 ---> 7 ---> 3 ]
4       10         20            [ 1 ---> 5 ---> 10 ---> 4 ]
5       5         6            [ 1 ---> 5 ]
6       2         22            [ 1 ---> 7 ---> 2 ---> 6 ]
7       7         7            [ 1 ---> 7 ]
8       3         12            [ 1 ---> 7 ---> 3 ---> 8 ]
9       12         20            [ 1 ---> 7 ---> 12 ---> 9 ]
10       5         9            [ 1 ---> 5 ---> 10 ]
11       8         28            [ 1 ---> 7 ---> 3 ---> 8 ---> 11 ]
12       7         15            [ 1 ---> 7 ---> 12 ]
13       6         26            [ 1 ---> 7 ---> 2 ---> 6 ---> 13 ]
Time taken by dvrp: 321 nanoseconds
```

DVRP for all nodes:

```
0       10         12            [ 12 ---> 10 ]
1       8         28            [ 12 ---> 3 ---> 8 ---> 11 ]
2       12         0            [ 12 ]
3       9         12            [ 12 ---> 9 ---> 13 ]
Dest Next Hop Dist Shortest Path

------------------------------------------------------------------------
-----

        7         26            [ 13 ---> 6 ---> 2 ---> 7 ---> 1 ]
        6         6            [ 13 ---> 6 ---> 2 ]
        12         20            [ 13 ---> 9 ---> 12 ---> 3 ]
        6         23            [ 13 ---> 6 ---> 4 ]
        10         27            [ 13 ---> 9 ---> 12 ---> 10 ---> 5 ]
        6         4            [ 13 ---> 6 ]
        2         19            [ 13 ---> 6 ---> 2 ---> 7 ]
        6         21            [ 13 ---> 6 ---> 8 ]
        9         7            [ 13 ---> 9 ]
10       12         24            [ 13 ---> 9 ---> 12 ---> 10 ]
11       6         29            [ 13 ---> 6 ---> 11 ]
12       9         12            [ 13 ---> 9 ---> 12 ]
13       13         0            [ 13 ]
Time taken by dvrp: 237 nanoseconds
```

Time Checking(nano seconds):

Before eliminating edge 4-10

Source	LSRP	DVRP
1	1.17689e+06	217
2	1.07665e+06	204
3	546260	306
4	252678	250
5	248706	160
6	5.34375e+06	237
7	335254	240
8	769846	253
9	237830	210
10	325950	201
11	338380	302
12	222629	208
13	230430	228

Removing edge 4-10:

```
remove 4-10
show
u|v | 1 2 3 4 5 6 7 8 9 10 11 12 13
-----------------------------------------------------
1 | 0 -1 -1 -1 6 -1 7 -1 -1 -1 -1 -1 -1
2 | -1 0 -1 -1 -1 2 13 -1 -1 -1 -1 -1 -1
3 | -1 -1 0 -1 -1 -1 1 4 -1 -1 -1 8 -1
4 | -1 -1 -1 0 -1 19 -1 -1 -1 -1 -1 -1 -1
5 | 6 -1 -1 -1 0 -1 -1 -1 -1 3 -1 -1 -1
6 | -1 2 -1 19 -1 0 -1 17 -1 -1 25 -1 4
7 | 7 13 1 -1 -1 -1 0 -1 -1 -1 -1 8 -1
8 | -1 -1 4 -1 -1 17 -1 0 -1 -1 16 -1 -1
9 | -1 -1 -1 -1 -1 -1 -1 -1 0 -1 -1 5 7
10 | -1 -1 -1 -1 3 -1 -1 -1 -1 0 -1 12 -1
11 | -1 -1 -1 -1 -1 25 -1 16 -1 -1 0 -1 -1
12 | -1 -1 8 -1 -1 -1 8 -1 5 12 -1 0 -1
13 | -1 -1 -1 -1 -1 4 -1 -1 7 -1 -1 -1 0
```

As we can observe edge 4-10 had been eliminated and changed from to -1.

Source	LSRP	DVRP
1	845642	214
2	944823	220
3	546260	231
4	4.18362e+06	203
5	316906	301
6	5.34375e+06	242
7	335254	256
8	1.69476e+06	292
9	606452	203
10	464406	219
11	406477	230
12	436815	256
13	385102	294

Bellman ford algorithm converges sooner and most of the time works better and omitting an edge doesn't make a big difference in the results.