NETCOMM LAB ASSESSMENT 3

Chirayu Rathi
19BCE2206

1. Implement Distance Vector Routing Protocol

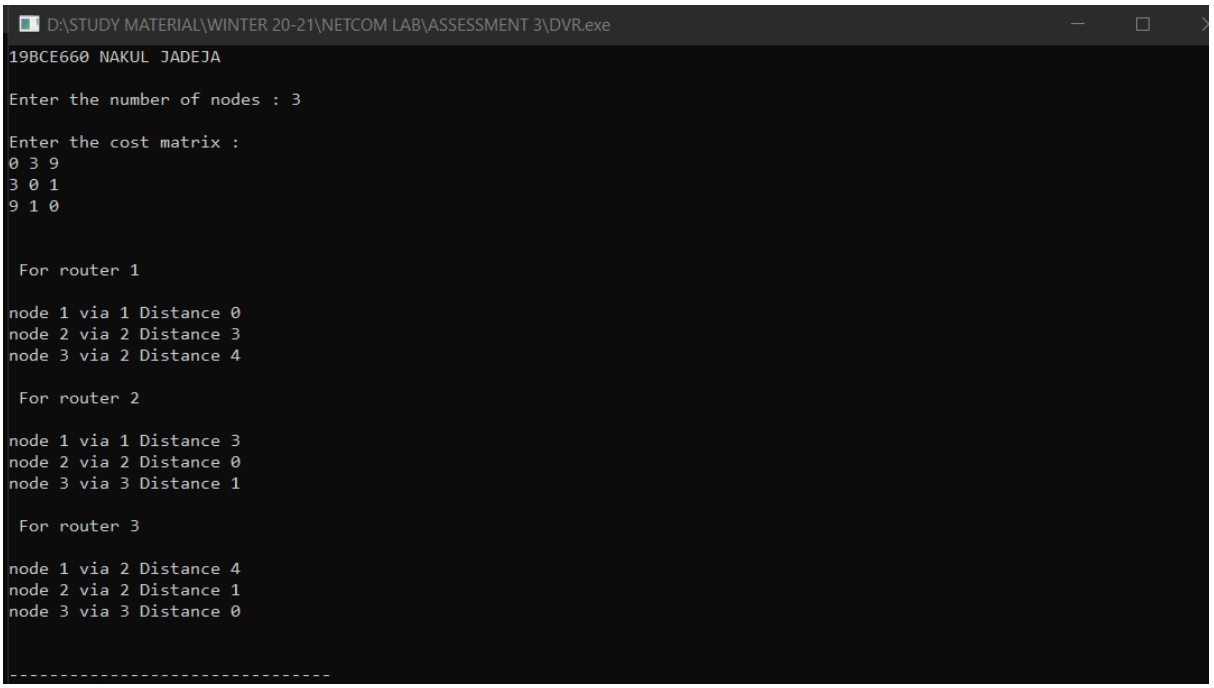
CODE:

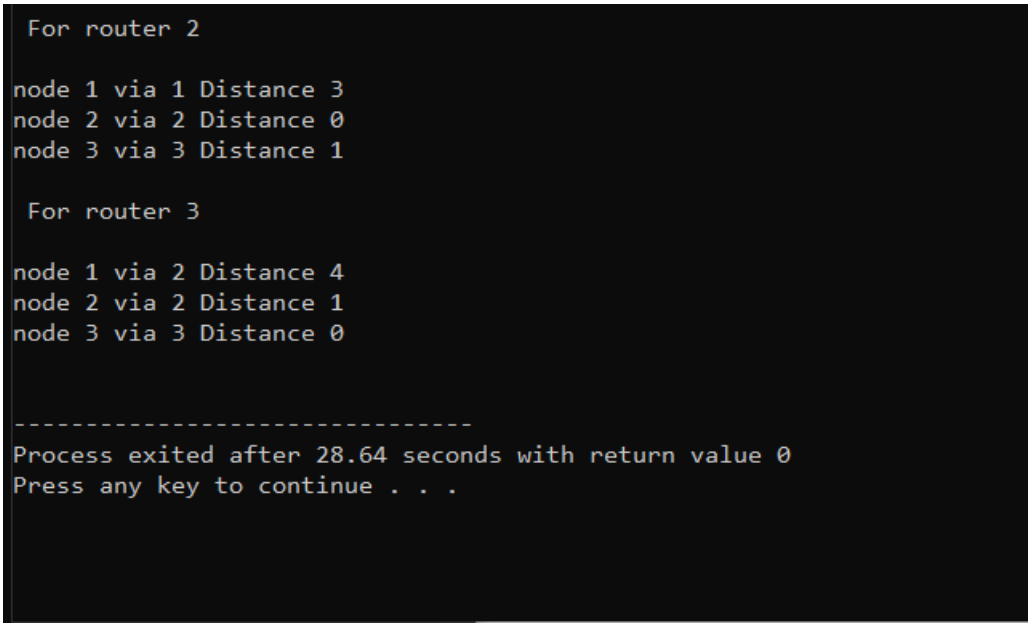
```
/*
Distance Vector Routing in this program is implemented using Bellman Ford Algorithm:-
*/
#include<stdio.h
>struct node
{
   unsigned dist[20];
   unsigned from[20];
}rt[10];
```

```
int main()
{
   int  costmat[20][20];
   int nodes,i,j,k,count=0;
   printf("19BCE660 NAKUL JADEJA \n");
   printf("\nEnter the number of nodes : ");
   scanf("%d",&nodes);//Enter the nodes
   printf("\nEnter the cost matrix :\n");
   for(i=0;i<nodes;i++)
   {
      for(j=0;j<nodes;j++)
      {

         scanf("%d",&costmat[i]
         [j]);costmat[i][i]=0;
         rt[i].dist[j]=costmat[i][j];//initialise the distance equal to
cost matrix
         rt[i].from[j]=j;

       }
      }

      do
      {
         count=0;
         for(i=0;i<nodes;i++)//We choose arbitary vertex k and
we calculate the direct distance from the node i to k using the cost
matrix
```

```
            //and add the distance from k to node
            jfor(j=0;j<nodes;j++)
            for(k=0;k<nodes;k++)
               if(rt[i].dist[j]>costmat[i][k]+rt[k].dist[j])
               {//We calculate the minimum
                  distancert[i].dist[j]=rt[i].dist[k]
                  +rt[k].dist[j]; rt[i].from[j]=k;
                  count++;
               }
         }while(count!=0);
         for(i=0;i<nodes;i++)
         {
            printf("\n\n For router
            %d\n",i+1);for(j=0;j<nodes;j++)
            {
               printf("\t\nnode %d via %d Distance
%d",j+1,rt[i].from[j]+1,rt[i].dist[j]);
            }
         }
      printf("\n\n");

   }
```

SCREENSHOT OF THE OUTPUT:





2. Implement Link State Routing Protocol

CODE:

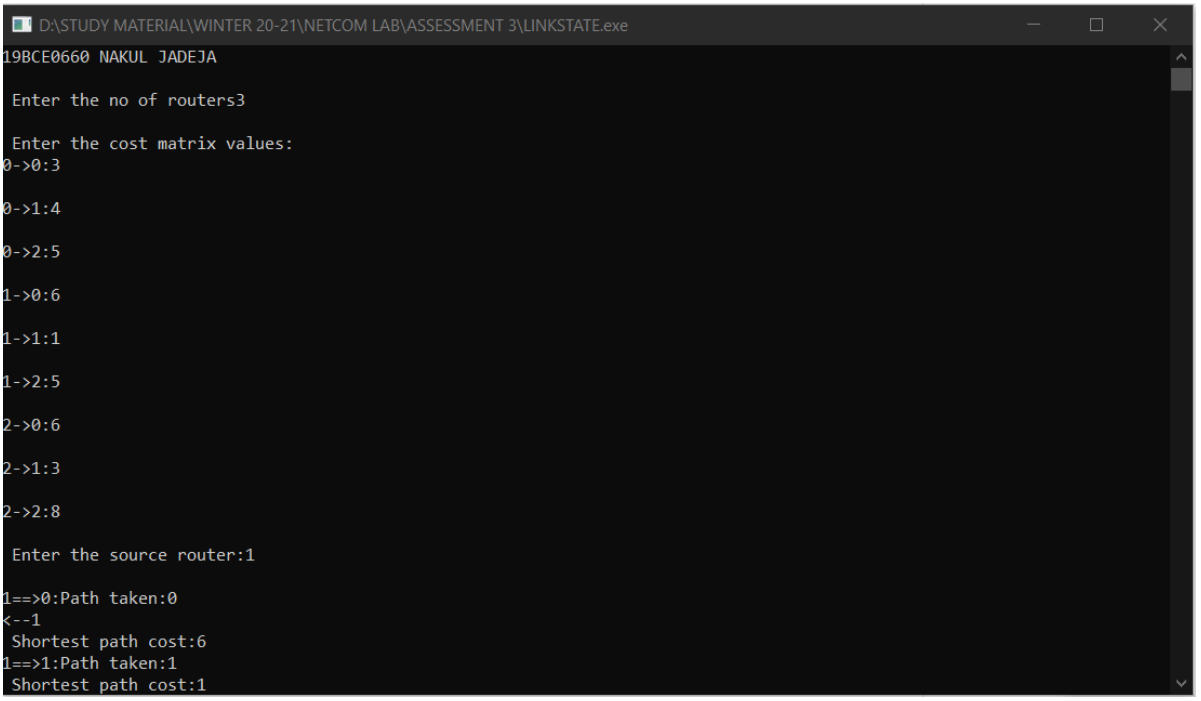
```
#include  <stdio.h>
# i n c l u d e
<s t r i n g . h > i n t
main()
{
```

```
int count,src_router,i,j,k,w,v,min;
int cost_matrix[100]
[100],dist[100],last[100];int flag[100];
        printf("19BCE0660 NAKUL JADEJA \n");
 printf("\n Enter the no of
routers");scanf("%d",&count);
printf("\n Enter the cost matrix values:");
for(i=0;i<count;i++)
{
for(j=0;j<count;j++)
{
 printf("\n%d->%d:",i,j);
scanf("%d",&cost_matrix[i][j]); if(cost_matrix[i][j]<0)cost_matrix[i]
[j]=1000;
}
}
 printf("\n Enter the source
router:");scanf("%d",&src_router);
for(v=0;v<count;v++)
{
flag[v]=0;
last[v]=src_router;
dist[v]=cost_matrix[src_router][v];
}
```

```
flag[src_router]=1;
for(i=0;i<count;i++)
{
min=1000;
for(w=0;w<count;w++)
{
if(!flag[w])
if(dist[w]<min)
{
v=w;
min=dist[w];
}
}
flag[v]=1;
for(w=0;w<count;w++)
{
if(!flag[w]) if(min+cost_matrix[v]
[w]<dist[w])
{
dist[w]=min+cost_matrix[v][w];
last[w]=v;
}
}
}
```

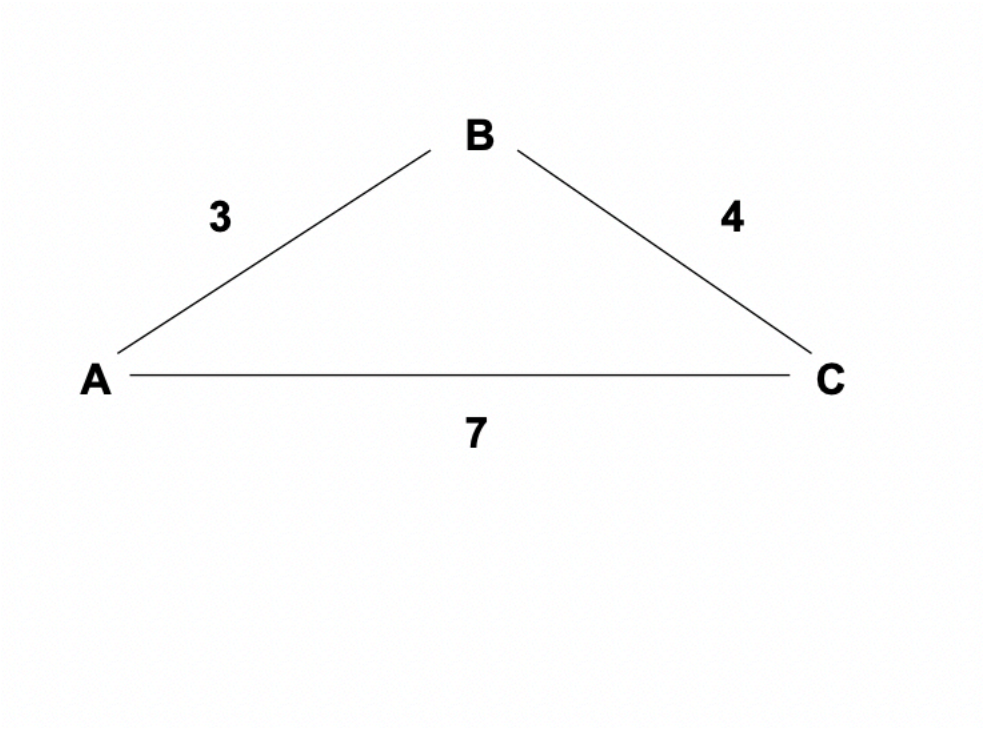
```
for(i=0;i<count;i++)
{
 printf("\n%d==>%d:Path taken:%d",src_router,i,i);w=i;
while(w!=src_router)
{
 printf("\n<--%d",last[w]);w=last[w];
}
 printf("\n Shortest path cost:%d",dist[i]);
}
}
```

SCREENSHOT OF THE OUTPUT:



```
 Enter the source router:1

1==>0:Path taken:0
<--1
 Shortest path cost:6
1==>1:Path taken:1
 Shortest path cost:1
1==>2:Path taken:2
<--1
 Shortest path cost:5
--------------------------------
Process exited after 57.95 seconds with return value 0
Press any key to continue . . .
```

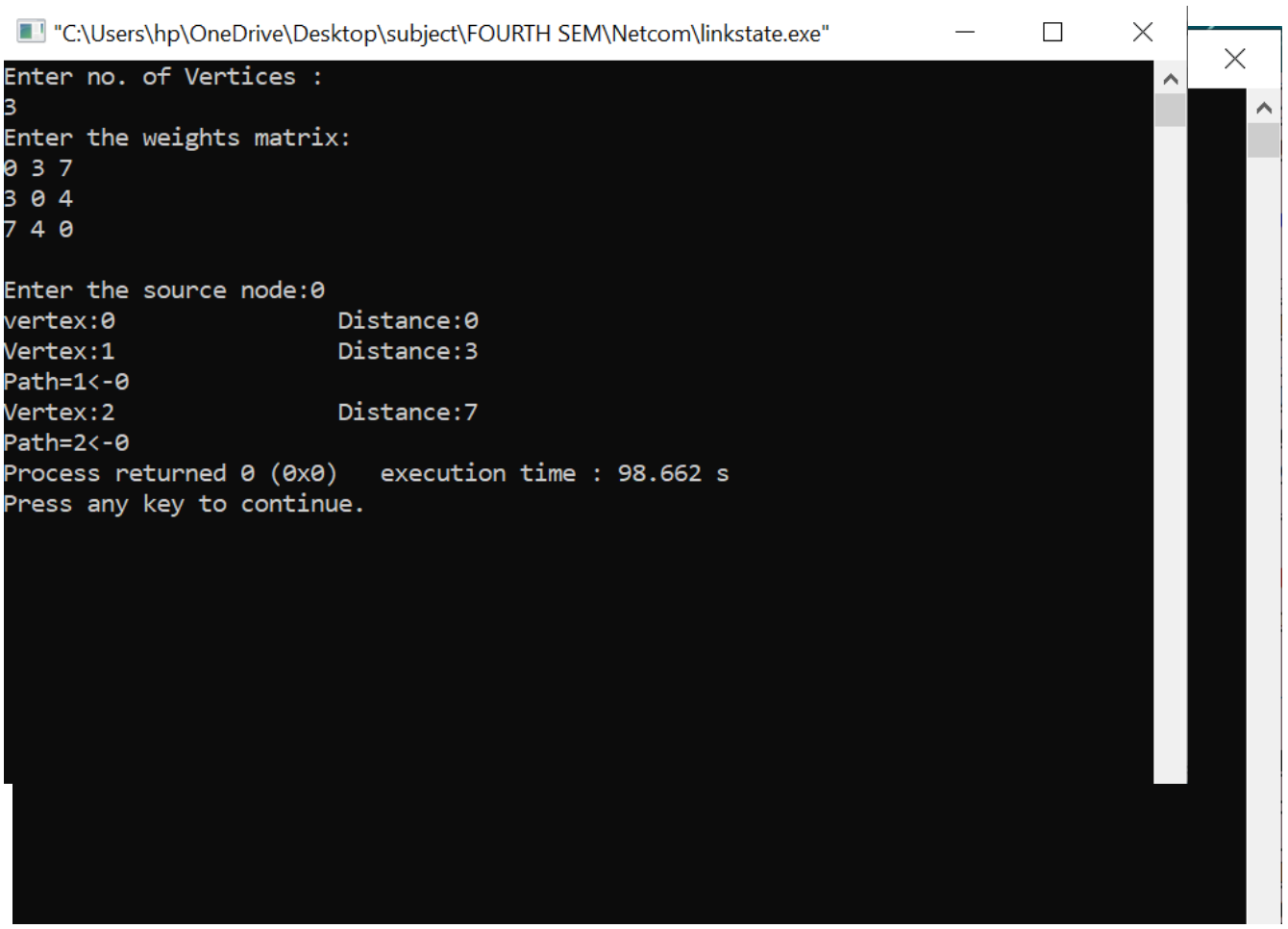
3. Show the Performance Evaluation between Distance Vector Routing Protocol and Link State Routing Protocol



ANSWER:

Distance Vector Output:

Link State routing output:



It is observed from the above example that distance vector routing updates and print all thelink states for each router while in Link state routing entire routing table is updated once for a given source node i.e. distance to all other vertices is found.

- Bellman-Ford algorithm is used for performing distance vector routing whereas Dijsktra isused for performing the link state routing.
- Time complexity of distance vector routing:O(n)Time complexity of link state routing:O(n^2)
- In distance vector routing the routers receive the information from the neighbour point ofview. On the contrary, in link state routing the router receive complete information on thenetwork topology.
- Distance vector routing calculates the best route based on the distance (fewest number ofhops). As against, Link state routing calculates the best route on the basis of least cost.
- Link state routing updates only the link state while Distance vector routing updates fullrouting table.
- The frequency of update in both routing technique is different distance vector updateperiodically whereas link state update frequency employs triggered updates.
- The utilization of CPU and memory in distance vector routing is lower than the link staterouting.

- The distance vector routing is simple to implement and manage. In contrast, the link state routing is complex and requires trained network administrator.
- The convergence time in distance vector routing is slow, and it usually suffers from count to infinity problem. Conversely, the convergence time in link state routing is fast, and it is morereliable.
- Distance vector doesn't have hierarchical structure while in link state routing the nodes canhave a hierarchical structure.

In distance vector routing the routing share, the information of the entire autonomous system and the information is shared only with neighbours. On the other hand, in link state routing therouters share the knowledge only about their neighbours and the information is shared with allrouters.