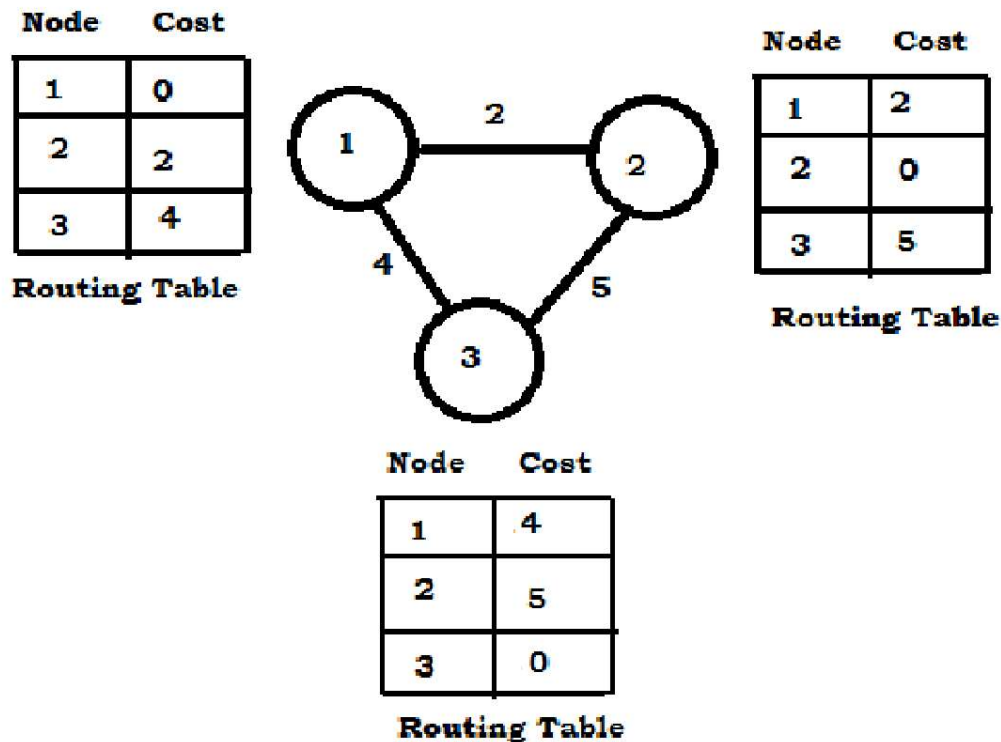


# **COMPUTER NETWORKS LAB**

## EXPERIMENT-6

**AIM:** Implement distance vector routing algorithm for obtaining routing tables at each node.

**PROGRAM:**



**HARDWARE REQUIREMENTS:** Intel based Desktop PC:- RAM of 512 MB

**SOFTWARE REQUIREMENTS:** Turbo C / Borland C.

### THEORY:

Distance Vector Routing Algorithms calculate a best route to reach a destination based solely on distance. E.g. RIP. RIP calculates the reach ability based on hop count. It's different from link state algorithms which consider some other factors like bandwidth and other metrics to reach a destination. Distance vector routing algorithms are not preferable for complex networks and take longer to converge.

### ALGORITHM/FLOWCHART:

Begin

**Step1:** Create struct node unsigned dist[20], unsigned from[20], rt[10]

**Step2:** initialize int dmat[20][20], n, i, j, k, count=0,

**Step3:** write "the number of nodes "

```

Step4: read the number of nodes "n"
Step5: write" the cost matrix :"  

Step6: initialize i=0  

Step7: repeat until i<n  

Step8: increment i  

Step9: initialize j=0  

Step10: repeat Step(10-16)until j<n  

Step11: increment j  

Step12:read dmat[i][j]  

Step13:initialize dmat[i][j]=0  

Step14:initialize rt[i].dist[j]=dmat[i][j]  

Step15:initialize rt[i].from[j]=j  

Step16:end  

Step17:start do loop Step (17-33)until  

Step18:intilialize count =0  

Step19:initialize i=0  

Step20:repeat until i<n  

Step21:increment i  

Step22:initialize j=0  

Step23:repeat until j<n  

Step24:increment j  

Step25:initialize k=0  

Step26:repeat until k<n  

Step27:increment k  

Step28:if repeat Step(28-32) until rt[i].dist[j]>dmat[i][k]+rt[k].dist[j]  

Step29:intialize rt[i].dist[j]=rt[i].dist[k]+rt[k].dist[j]  

Step30:intialize rt[i].from[j]=k;  

Step31:increment count  

Step32:end if  

Step33:end do stmt  

Step34:while (count!=0)  

Step35:initialize i=0  

Step36:repeat Steps(36-44)until i<n  

Step37:increment i  

Step38:write ' state values for router',i+1  

Step39:initialize j=0  

Step40:repeat Steps ( 40-43)until j<n  

Step41:increment j  

Step42:write 'node %d via %d distance % ',j+1,rt[i].from[j]+1,rt[i].dist[j]  

Step43:end  

Step44:end  

end

```

#### **SOURCE CODE:**

```

#include<stdio.h>
#include<conio.h>

```

```

struct node
{
unsigned dist[20];
unsigned from[20];
}rt[10];
int main()
{
int dmat[20][20];
int n,i,j,k,count=0;
clrscr();
printf("\n Enter the number of nodes : ");
scanf("%d",&n);
printf("Enter the cost matrix :\n");
for(i=0;i<n;i++)
for(j=0;j<n;j++)
{
scanf("%d",&dmat[i][j]);
dmat[i][i]=0;
rt[i].dist[j]=dmat[i][j];
rt[i].from[j]=j;
}
do
{
count=0;
for(i=0;i<n;i++)
for(j=0;j<n;j++)
for(k=0;k<n;k++)
if(rt[i].dist[j]>dmat[i][k]+rt[k].dist[j])
{
rt[i].dist[j]=rt[i].dist[k]+rt[k].dist[j];
rt[i].from[j]=k;
count++;
}
}while(count!=0);
for(i=0;i<n;i++)
{
printf("\nState value for router %d is \n",i+1);
for(j=0;j<n;j++)
{
printf("\nnode %d via %d Distance%d",j+1,rt[i].from[j]+1,rt[i].dist[j]);
}
}
printf("\n");
}

```

OUTPUT:

```
Turbo C++ IDE

Enter the number of nodes : 3
Enter the cost matrix :
0 2 4
2 0 5
4 5 0

State value for router 1 is
node 1 via 1 Distance0
node 2 via 2 Distance2
node 3 via 3 Distance4
State value for router 2 is
node 1 via 1 Distance2
node 2 via 2 Distance0
node 3 via 3 Distance5
State value for router 3 is
node 1 via 1 Distance4
node 2 via 2 Distance5
node 3 via 3 Distance0
```