International Journal of Research Studies in Computer Science and Engineering (IJRSCSE)
Volume. 1, Issue 5, September 2014, PP 21-29
ISSN 2349-4840 (Print) & ISSN 2349-4859 (Online)
www.arcjournals.org

Adaptive Routing Algorithms in Distributed Networks

K.Rangaswamy[1] & G.Sreenivasula Reddy[2]

[1]Assistant professor, Department of CSE, CBIT, Proddatur, Kadapa (dist), A.P
[2]Professor, Department of CSE, VBIT, Proddatur, Kadapa (dist), A.P

Abstract: *Routing is an important function to run some important operations in the computer networks, it influence both the process of managing the network as the quality of services in world wide networks. The management of transferring the message has to fulfill the requirements volume of traffic to be transmitted as to prevent congestions for reducing the transmission delays. These two requirements in general are contradictory. The most favorable process of sending the messages is a key issue for the quality of the information services. Routing in networks, applying shortest path algorithms widely used in communication protocols in WIDE AREA NETWORK.*

Keywords: *Routing, Network traffic, Optimization, shortest path algorithm computer networks*

1. INTRODUCTION

The main function of the network layer is routing packets from source machine to destination machine. The algorithms that choose the routes and the data structures .ROUTING ALGORITHM is the part of network layer software responsibility is to decide that which output line an incoming packet should be transmitted on. The efficiency of a routing algorithm depends on its performance, during congestions in the network it must perform route choice and delivery of messages. The performance of the routing is estimated depends on the throughput in the network and quality of the service (Packet delay).The routing influences the flow control with the delays on links

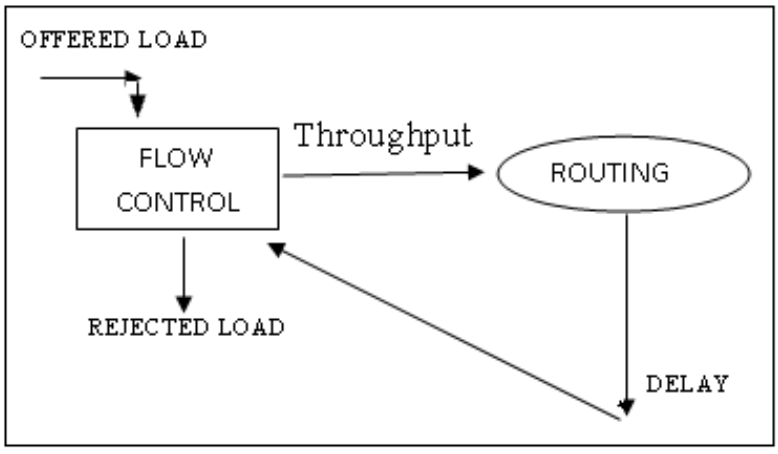


Fig: *Interaction between flow control and routing from the figure, when the throughput increases then delay time also increases.*

2. TYPES OF ROUTING ALGORITHMS

2.1. Optimal Routing

In the communication systems, the routers compute the flow transmissions according to the shortest path algorithm. Optimal routing algorithm is efficient in finding optimal route, according to the link weights, presenting the traffic load on them. This optimal routing algorithm can't flow through alternative paths. Generally several paths are present between source and destination nodes in the common network structure. Commonly an **open shortest path first** (OSPF) protocol routes the path depends on their cost, but it does not estimate and apply alternative routing to

available paths. Thus QoS (Quality of Services) is not supported only by shortest path management.

Thus the average delay per packet is reduced also at steady or low traffic conditions.

This architecture insists routers to broadcast the the local topology information to all routers. The process of providing QoS in the routers to apply traffic prioritization. This idea is used to classify that the traffic to a multiple levels of priority queues. Those priorities are assigned on packet peculiarities the protocol uses packet type, source and destination networks. Enhancements done by subdividing the link capacity into different classes finally the traffic is assigned to each class and the routers serve each class with not the same priority.

The optimization problem for this case can be stated as:

$$\max q$$

$$\sum_{j \in FS(i)} x_{ij} + \sum_{j \in RS(i)} x_{ji} = \begin{cases} q, & if \quad i = s \\ -q, & if \quad i = t \\ 0 & if \quad otherwise \end{cases}$$

subject to

$$\sum_{i,j \in A} \sum_{i,j \in A} d_{ij}(x_{ij}) \le T^*$$

$$0 \le x_{ij} \le c_{ij} \quad \forall (i,j) \in A,$$

$$x - \text{int } eger$$

where FS(*i*) and RS(*i*) denote the forward and reverse links for node *i*, A is a set of arks, *cij* is the capacity of link (*i, j*), *q* is the flow to be routed from source node s to destination node *t*, *T** is the maximal feasible delay, and *dij* is the delay function for the link(*i, j*). Routers use "Routing algorithms" to find the best route to destination .considering parameters like the number of Hops (a hop is the trip a data packet takes from one router or intermediate point to another in the network), time delay and communication cost of packet transmission.

There exist two important routing algorithms for gathering the information: “Global routing algorithms" and "Decentralized routing algorithms". In global routing algorithm every node has information about the remaining nodes in the network and speed of the message passing time. These algorithms are called as “link state algorithm”

2.2. Link State Algorithm

The main principles of link state algorithm Each router keeps a topology database of whole network link state updates flooded, or multicast to all network routers compute their routing tables based on topology often uses Dijkstra’s shortest path algorithm.

It is mainly used in the OSPF.

Mainly it consists of several operations in link state algorithm:

Step1:

Finding of routers which are physically connected to the routers and also its IP address**.** When router starts working it will send the “HELLO” packet over the network. All routers with in the network will receive the message its replays the ip address of that particular router.

Step2:

Delay time for the neighboring routers in the network will be measured. Routers will send the Echo packets over the network, every router that receives these packets replies with an Echo reply Packet. By dividing the Round Trip Time by 2, routers can count the delay time. The Round Trip Time is a measure of the current delay on a network, found by timing a packet bounced off some remote host. This time includes the time in which the packets reach the destination and the time in which the receiver processes it and replies.

Step3:

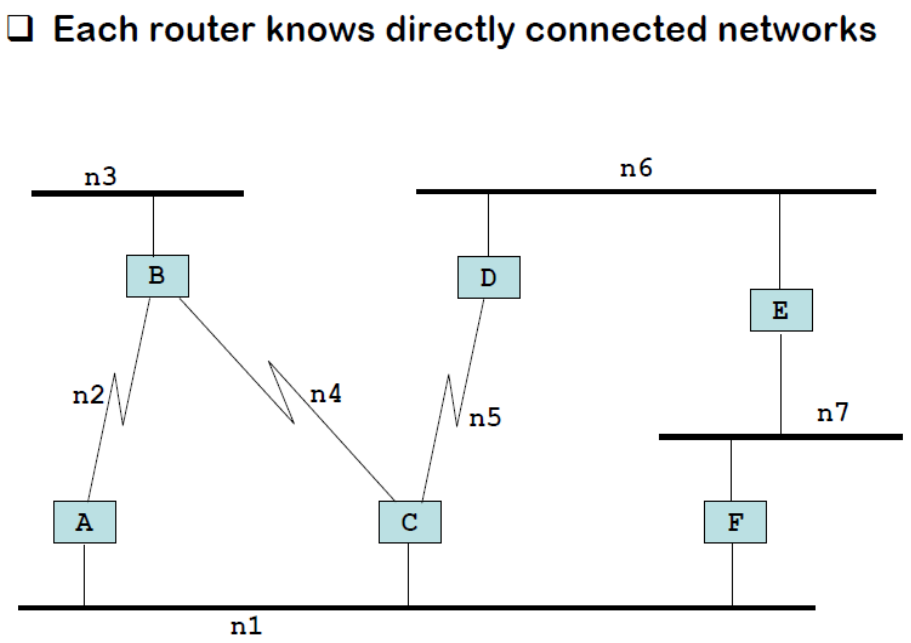
Router will broadcast all its information over the network for other routers and receives .Thus all routers share their knowledge and broadcast their information to each other and each router is acquainted with the structure and the status of the network.

Step4:

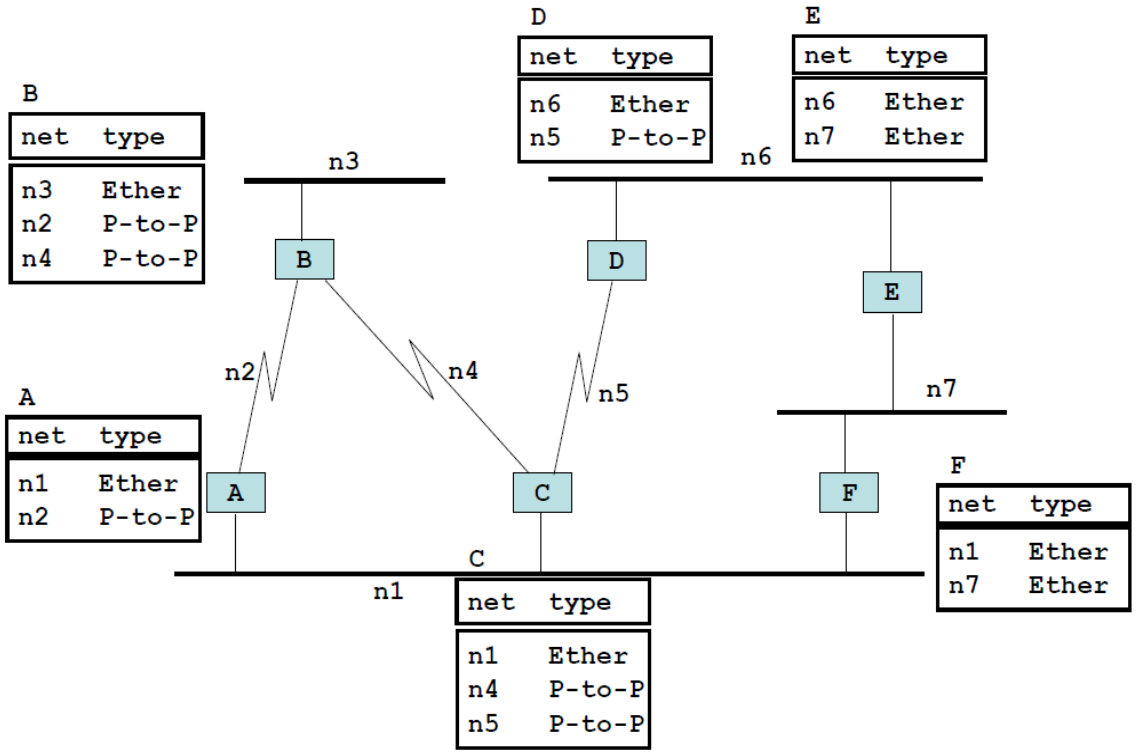
The router will evaluates the best route between two nodes of network**.** Thus the best route for the packets to every node is chooses. For this evaluation the shortest path algorithm of Dijkstra is performed.

Flooding topology in link state

Neighbouring nodes synchronize before starting any relationship Hello protocol; keep alive initial synchronization of database description of all links (no information yet) Once synchronized, a node accepts link state advertisements contain a sequence number, stored with record in the database only messages with new sequence number are accepted accepted messages are flooded to all neighbors sequence number prevents anomalies (loops or black holes).

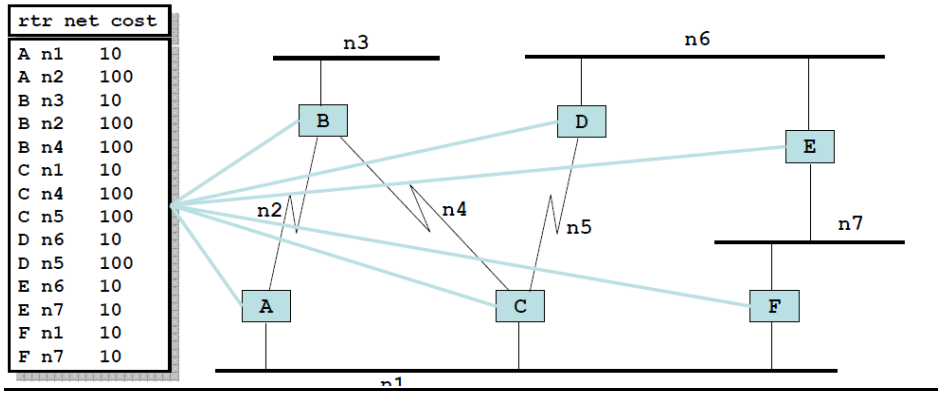


Intial routing tables



After Flooding

- The local metric information is flooded to all routers
- After convergence, all routers have the same information



2.3. Dijkstra Shortest Path Algorithm

Main Aim: Find the least cost paths from a given node to all other nodes in the network.

Notation:

Dij = Link cost from i to j if i and j are connected

Dn = Total path cost from s to n

M = Set of nodes so far for which the least cost path is known

Method:

Initialize: M={s}, Dn = dsn

Find node w$\notin$ M, whose Dn is minimum

Update Dn

One of the oldest and best known problem in the field of distributed algorithms is to compute shortest paths between nodes in a network. This problem arises in the following context. We have a network of links and nodes (processors). Each link (I,J) is characterized by a direction dependent length LEN(I,J) that can change with time and can only be observed at node I. The nodes execute a distributed algorithm to keep track of the shortest distances between themselves and the other nodes, in the course of which they communicate with adjacent nodes by transmitting messages over the links. The most popular solution to this problem is the Ford-Bellman method that was originally introduced in the Arpanet and is now used in a large number of networks.

The following is a simple set of instructions that enables to follow the algorithm:

S1: Label the start vertex as 0.

S2: Box this number (permanent label).

S3: Label each vertex that is connected to the start vertex with its distance (temporary label).

S4: Box the smallest number.

S5: From this vertex, consider the distance to each connected vertex.

S6: If a distance is less than a distance already at this vertex, cross out this distance and write in the new distance. If there was no distance at the vertex, write down the new distance.

S7: Repeat from step 4 until the destination vertex is boxed.

Before evaluating the complexity of the algorithm, we must precise when COMPUTE () is executed after a Triggering. There are two traditional possibilities, and we also suggest another :A) event driven: run COMPUTE () whenever a topology change occurs or an update message is received.

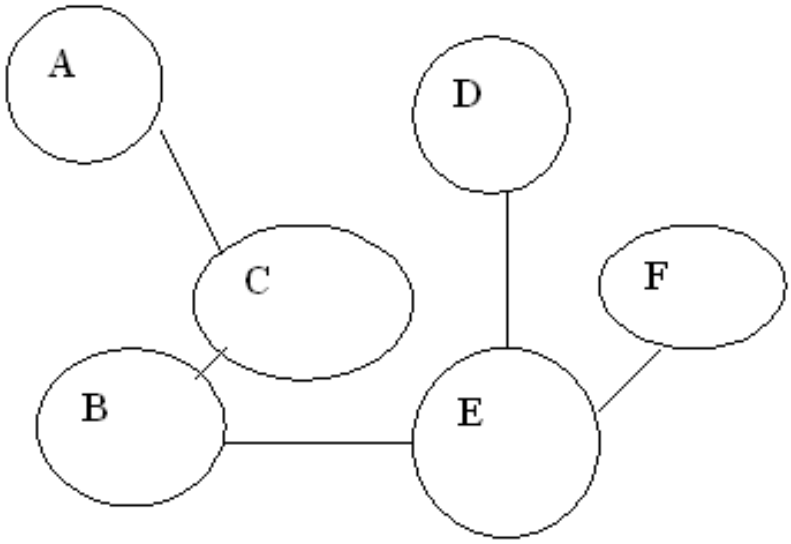
One expects that this would be the fastest. if the output links have finite capacity this might Result in update messages queuing for transmission.

B) Periodic: run COMPUTE () at each node on a periodic basis, the periods need not be the same at all

Dijkstra algorithm in c as follows

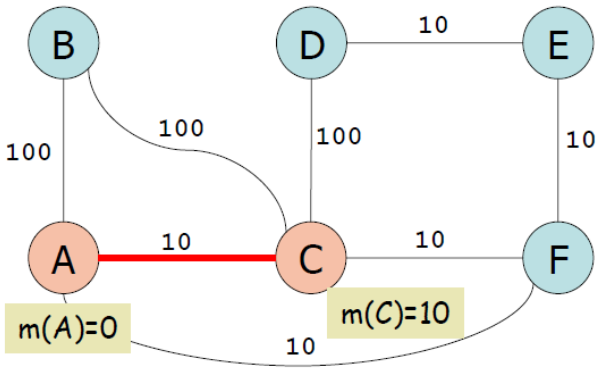
```
#include<stdio.h>
#include<conio.h>
Int min_cost(int *dist,int *visit,int n);
{
Int I,min=9999,index;
For(i=1;i<=n;i++)
If(dist[i]<min&& visit[i]==0)
{
Min=dist[i];
Index=I;
}
Return index;
}main()
{
int cost[50][50],n,I,j,src;
Int *dist,*visit,k;
Clrscr();
Dist=(int *)calloc(50,size of(int));
visit=(int *)calloc(50,size of(int));
printf("\n group consists of how many nodes");
scanf("%d",&n);
printf("\n enter %d ele of cost adjacency
matrix",n*n);
for(i=1;i<=n;i++)
for(j=1;j<n;j++)
{
Scanf("%d:,cost[i][j]);
If(cost[i][j]==0)
Cost[i][j]=9999;
Visit[i]=0;
}
Printf("\n enter src node: \n");
Scanf("%d",&source);
For(i=1;i<=n;i++)
Dist[i]=cost[source][i];
Visit[source]=1;
                    For(i=2;i<n;i++)
{
K=min_cost(dist,visit,n);
Visit[k]=1;
For(j=1;j<n;j++)
If(cost[k][j]!=9999 && visit[j]=0)
If(dist[k]+cost[k][j]<dist[j])
{
Dist[j]=dist[k]+cost[k][j];

            Printf("\n %d-%d:%d-%d-%d",src,j,k,j)
}
}
For(i=1;i<n;i++)
If(i!=source)
Printf("\n cost from %d is %d ",
source,I,dist[i]);
Getch();
}
```

Example for Dijkstra Algorithm



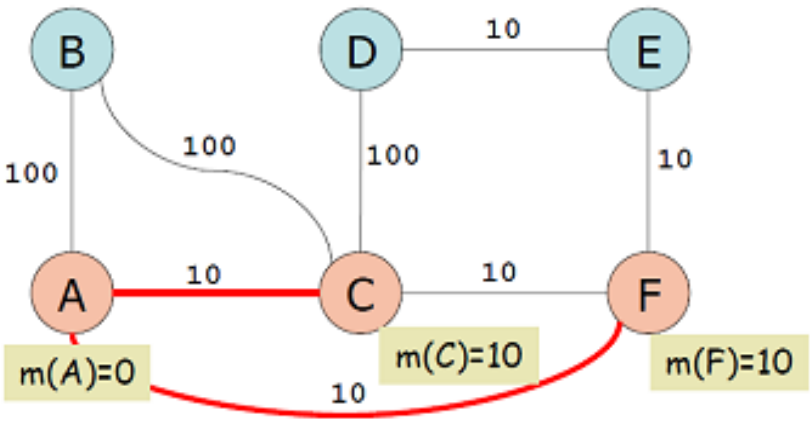
Resulting Forwarding Table in C

destination	link
A	(C,A)
B	(C,B)
D	(C,B)
E	(C,B)
F	(C,B)

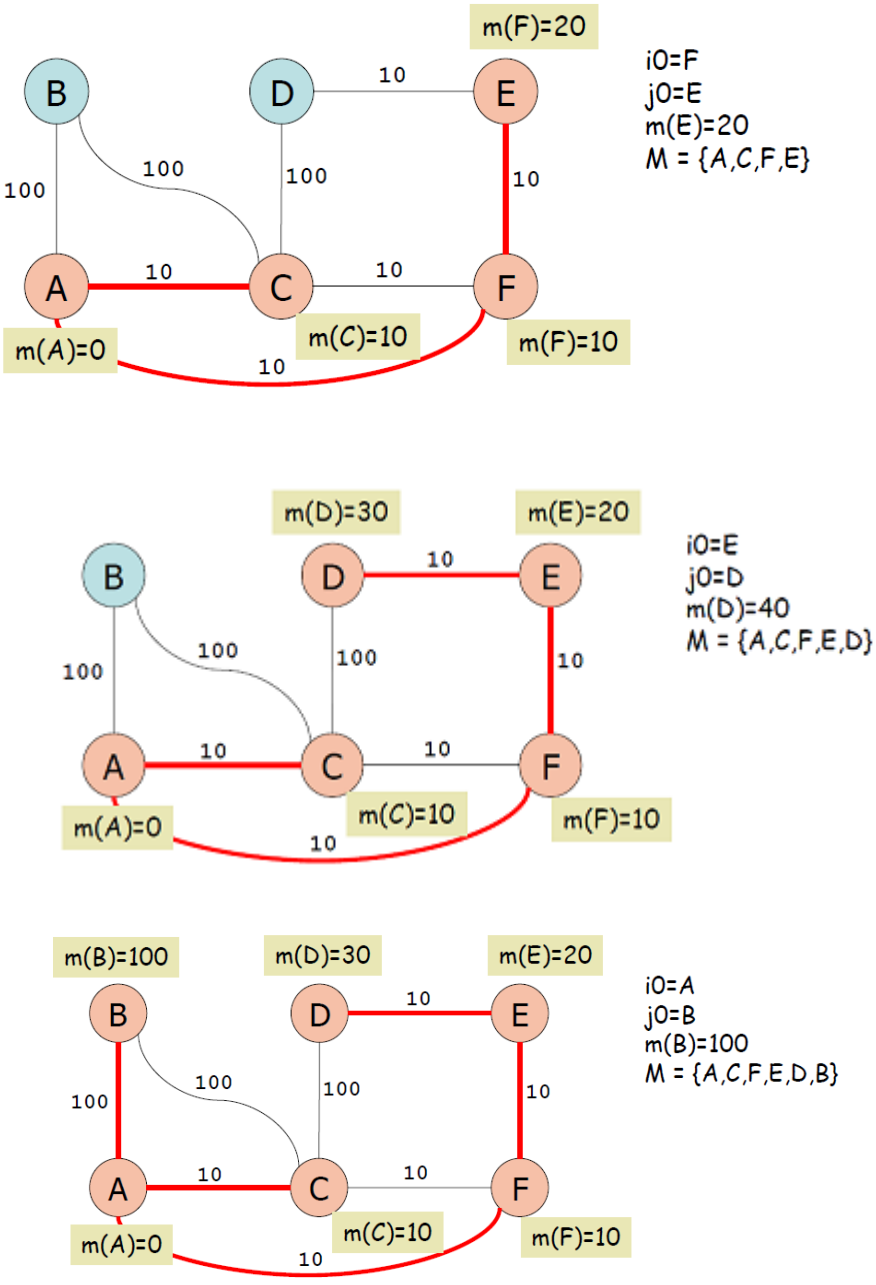


init: M = { A }

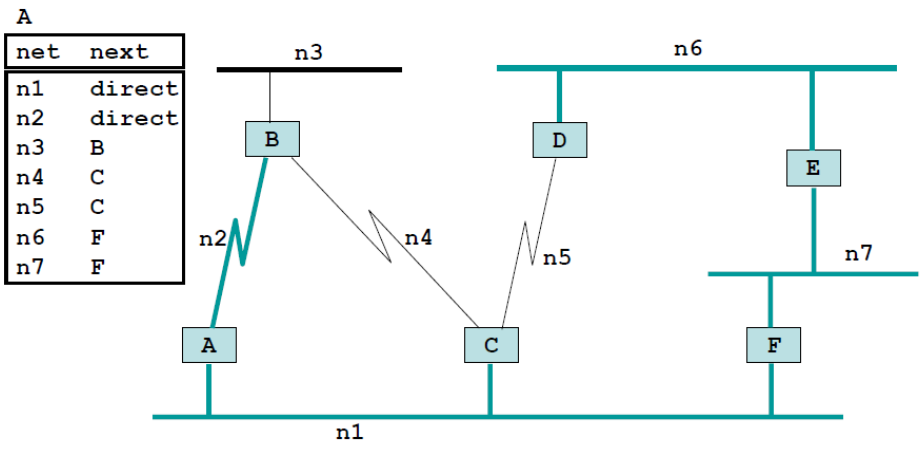
step 1:
i0=A
j0=C
m(C)=10
M = {A, C}



i0=A
j0=F
m(F)=10
M = {A,C,F}



Routing table of A



Another Example for the dijkstra algorithm by following above C-program:

Let us consider:

Graph consists of 4 nodes

Enter 16 elements in adjacency matrix:

0 8 7 0

8 0 0 9

7 0 0 6

0 9 6 0

Enter the source code:2

Cost from 2 to 1 is ‘8’.

Cost from 2 to 3 is ‘5’.

Cost from 2 to 4 is ‘9’.

3. CONCLUSION

By this routing algorithms we conclude that the For providing of quality by implementing the multiple routes. In moving of routing path oscillations must be avoided but sensitivity to congestions may be significant. In centralized routing systems failure of traffic management centre may be harm for the system. For avoiding the congested routes the information has to update continuously in the adaptive routing. The three most important performance measures Are quality and quantity of the service and speed.

REFERENCES

[1] Computer network, S.Tanenbaum, fourth edition, Pearson education.

[2] Bertsekas, D., R. Gallager. Data Networks (2nd ed), Prentice Hall, Englewood Cliffs, NJ, 1992.

[3] Brakmo L. , L. Peterson. TCP Vegas: End to End Congestion Avoidance on a Global Internet. IEEE Journal of Selected Areas in Communications, Vol. 13, No. 8, pp. 1465–1480, Oct. 1995.

[4] Chao, H., , C. Lam, E. Oki, Broadband Packet Switching Technologies—A Practical Guide to ATM Switches and IP Routers, John Wiley & Sons, 2001.

[5] M. Christiansen, M., K. Jeffay, D. Ott, F. D. Smith, Tuning Red for Web Traffic, IEEE/ACM Transactions on Networking, Vol. 9, No. 3 (June 2001), pp. 249–264, http://www.cs.unc.edu/~jeffay/papers/IEEE-ToN-01.pdf

[6] DeClerc, J., O. Paridaens, Scalability Implications of Virtual Private Networks, IEEE Communications Magazine, 40(5), May 2002, pp. 151–157.

[7] M. Donahoo,M., K. Calvert, TCP/IP Sockets in C: Practical Guide for Programmers, Morgan Kaufman, 2000.

[8] S. Floyd, S., K. Fall, Promoting the Use of End-to-End Congestion Control in the Internet, IEEE/ACM Transactions on Networking, Vol. 6, No. 5 (Oct. 1998), pp. 458–472. http://www.icir.org/floyd/end2end-paper.html

[9] Floyd, S., A Report on Some Recent Developments in TCP Congestion Control, IEEE Communications Magazine, 2001, [10] Fortz,B., J.Rexford, M.Thorup. Traffic Engineering with Traditional IP Routing Protocols. IEEE Communication Magazine, 2002,

AUTHORS' BIOGRAPHY

K.Rangaswamy is currently an Assistant Professor of Computer Science and Engineering at Chaitanya Bharathi Institute of Technology Proddatur, Andrapradesh.His most focus on networks.He received M.Tech degree in Computer Science at Bharath University, Chennai in 2011, his B.Tech degree in Computer Science and Engineering from JNTUA Anatapuramu in 2009.

G.Sreenivasula Reddy has prosecuted his B.E[Mechanical Engineering] from Bangalore University in 1996 , M.C.A from Madurai Kamaraj University in 2002 and M.Tech degree in Computer Science Engineering in 2007 from RGMCET Nandyal [Affiliated to JNTU, Hyderabad]. He worked as an Assistant Professor from 2003 to 2009 in Vaagdevi Institute of Technology. Presently he is working as Professor and In-charge Principal , Vignana Bharathi Institute of Technology,Proddatur, Y.S.R. Dist, Andhra Pradesh. He got admitted into Ph.D course in S.V.University in 2010. His research area is DATA MINING and IMAGE PROCESSING.