Routing algorithms in computers networks

Todor Stoilov, Krasimira Stoilova

***Abstract**: Routing in computer network is an essential functionality, which influence both the network management as the quality of services in global networks. The management of the traffic flows has to satisfy requirements for volume of traffic to be transmitted as avoidance of congestions for decreasing the transmission delays. These two requirements in general are contradictory. The optimal traffic management is a key issue for the quality of the information services. Routing in networks, applying shortest path algorithm is widely used in communication protocols in WAN. Short explanations and illustration of these algorithms is given.*

***Key words**: optimization, routing, network traffic, shortest path algorithmq computer networks*

INTRODUCTION

The routing algorithm is described by [2] as network layer protocol that guides packets (information stored as small strings of bits) through the communication subset to their correct destinations. Some reasons for the complexity of routing algorithms are: coordination between the nodes in the network; failures of the links and nodes; congestion of traffic links. Two types of algorithms are used for routing in networks: shortest path routing algorithms and optimal routing based on other measures [1, 5, 9]. The efficiency of a routing algorithm depends on its performance, during congestions in the network. The routing algorithms must perform route choice and delivery of messages. The performance of the routing is assessed according to the throughput in the network (quantity of data transfer) and the average packet delay (quality of service). Following [2,3,8], the routing influences the flow control with the delays on links, fig. 1

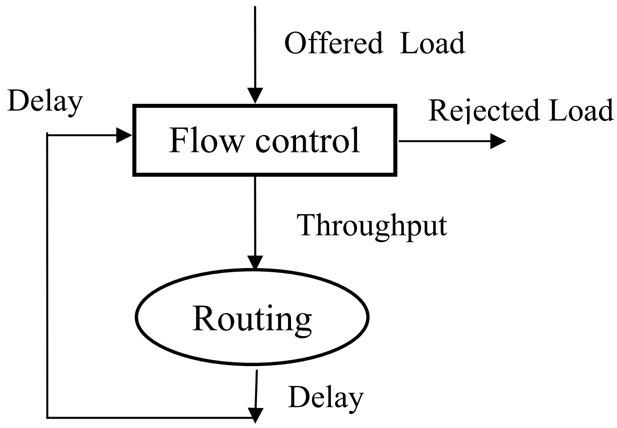

Fig.1 Interaction between flow control and routing

As it has been seen, the quantity of the transmission is in contradiction with the quality of service, because it influences with a delays by the feedbacks. Thus by increasing the throughput, the delays increase [3,11].

Optimal Routing

In Internet environment, the routers compute the flow transmissions according to the shortest path algorithm [10,12,13]. This algorithm is efficient in finding optimal route, according to the link weights, presenting the traffic load on them. The limitation of this algorithm is that it can not route the flow along alternative paths. In common network structure it exists always several paths between the source and destination nodes. Now the OSPF protocol routes according to the shortest path criteria, but it does not estimate and apply alternative routing to available paths. Thus Quality of Services (QoS) is not supported only by shortest path management.

The optimal routing under (QoS) requirements is a complex problem for implementation [4,6].

Such architecture insists routers to broadcast the local resource status and the local topology information to all routers. One manner of providing QoS in routers is to apply traffic prioritization. The idea is to classify the traffic to a multiple levels of priority queues. The priorities are assigned on packet peculiarities: the protocol uses packet type, source and destination networks. Enhancements are done by subdividing the link capacity into different classes. The traffic is assigned to each classes and the routers serve each class with different priority. However the traffic prioritization improves the QoS by class of traffic on a given link, but that link is chosen by the shortest path routing mechanism, which is independent of the QoS requirements.

The optimal routing algorithm must keep the delays low as the flow control increases. Thus the routing increases the throughput and restricts the delay for the packet, during high traffic conditions. Thus the average delay per packet is reduced also at steady or low traffic conditions. The optimization problem for this case can be stated as: .

max q

subject to

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

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

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

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

where FS(*i*) and RS(*i*) denote the forward and reverse links for node *i,* A is a set of arks, c_{ij} 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*).

This optimization problem is titled maximal flow probe; subject to the conservation constraints and additional time delay constraints. The solution of this problem has to find the best routing for the traffic flow and the calculations have to be performed by routers. A router is used to manage network traffic and finding the best route for packets. Hence the routers should have some information about network status to make decision about how to send packets. 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.

Based on the manner of how routers gather information about the network and the type of its utilization it exists two major routing algorithms: “Global routing algorithms" and "Decentralized routing algorithms". In Global Routing algorithms, every router must have complete information about other routers in the network and the traffic status. These algorithms are known as LS (Link State) algorithms. Reverse in Decentralized routing algorithms, each router has information about the routers that is directly connected to and not about all routers in network. These Algorithms are known as DV (Distance Vector) algorithms.

Sequence of operation of LS algorithm

Step 1: Identification of the routers, which are physically, connected to the router and estimation of their IP addresses**.** When a router starts working, first it sends a "HELLO" packet over network. All the routers that receive this packet reply with a message that contains their IP addresses.
Step2: The delay time for neighbour routers is measured. The routers send Echo packets over the network, every router that receives these packets replies with an Echo reply packet. By dividing 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: The router broadcast its information over a network for other routers and receives theirs. 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 network.
Step4: The router evaluates the best route between two nodes of network**.** Thus the best route for packets to every node is chosen. For this evaluation the shortest path algorithm of Dijkstra is performed. In this algorithm, router, based on the information from step1, builds a graph of network. This graph shows the location of the routers in the network and their links. Every link is labelled with a weight, titled as cost of link. This number is a function of delay time, average traffic or it is the number of hops between nodes.

Description of Dijkstra short path algorithm

The algorithm performs several rules:
Rule1: A graph of the network is built network and the adjacency matrix a [i, j] with the weight of links is defined. For the case when a direct link between node Vi and Vj is missing, the weight of the link is assumed as infinity. The source and the destination nodes are noted as NS and NT.
Rule2: A status record set is established for every node with three fields:

- The first field that shows the previous node, named "predecessor" field.
- The second filed is named "Length" field and it shows the sum of weights from source to that node.
- The last field, named "Label" filed, shows the status of the node. Each node can have one status mode: "Permanent" or "Tentative".

Rule3: Initialization of the status record set for all nodes and setting all “Length” to Infinity, and all "Label" as tentative.
Rule 4: Labeling node NS as t node and marking its "Label" as "Permanent". When a label changes to permanent, it never changes again. T node rules as a current chosen node.
Rule5: For all tentative nodes, directly linked to t node, status record set is updated.
Rule6: From all the tentative nodes, choose the one whose weight to NS is less and set it as t node.
Rule7: If this node is not the destination NT, then, go to step 5.
Rule8: If this node is NT, then extract its previous node from status record set and do this until return to NS. The nodes show the best route from NS to NV.

Example of using Dijkstra algorithm

The example presents the best route between A and E, fig.2. There are 6 possible routes between nodes A and E (ABE, ACE, ABDE, ACDE, ABDCE, ACDBE). Obviously ABDE is the best route because its weight is less than other routes. The application of the Dijkstra algorithm on this example is performed accordingly.

- The source node (A) is chosen as current node t and its label is marked as permanent. In fig.2 the permanent nodes are noted in black, and the t node is marked by arrow.
- The states of nodes B and C, which have direct link to the current node t are changed. Because node B has less weight, its state is changed to permanent and the label t goes on it, fig.3.

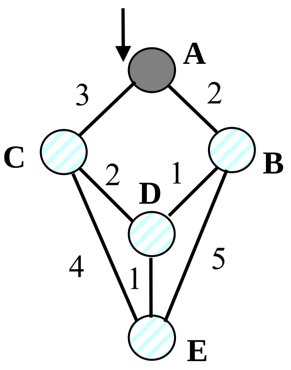

Fig.2 Initial state of Dijkstra algorithm

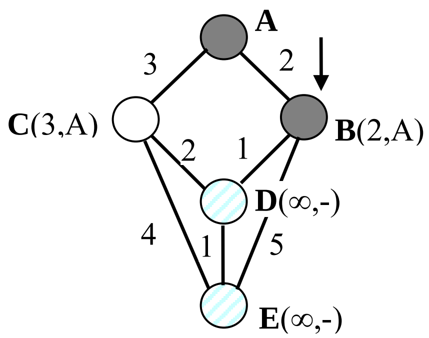

Fig. 3. Choice of **B** as current node t

- Following the rules for changing the states of the nodes D and E, D has less weight and it is chosen as the next current node t with permanent state, fig.5.

-The final rule of the algorithm arrive directly to the destination node E, fig5.

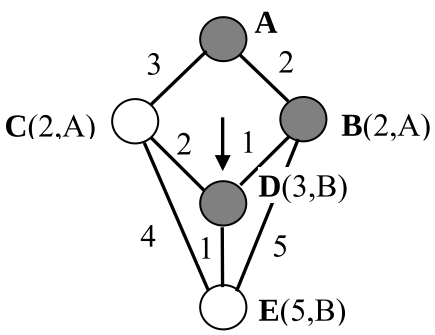

Fig. 4 The current state is on node D

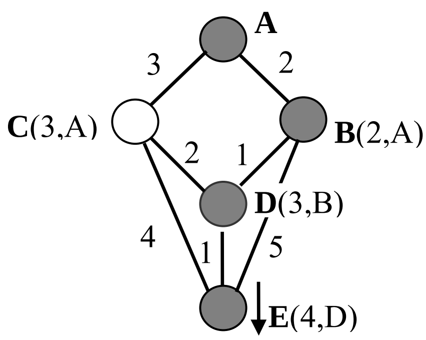

Fig. 5 The current state is on node E

- The identification of the shortest route is performed, according to the records in each status node record for the "predecessors" field. Following the reverse records, the optimal route is identified as the path A-B-D-E. The total weight of this route is 2+1+1=4.

An example of C code for the Dijkstra algorithm is given in fig.6, [7].

```
#define MAX_NODES 1024                /* maximum number of nodes */
#define INFINITY 1000000000           /* a number larger than every maximum path */
int n,dist[MAX_NODES][MAX_NODES];  /*dist[I][j] is the distance from i to j */
void shortest_path(int s,int t,int path[ ])
{struct state {                                  /* the path being worked on */
int predecessor ;                                /*previous node */
int length                                       /*length from source to this node*/
enum {permanent, tentative} label                /*label state*/
}state[MAX_NODES];
int I, k, min;
struct state *
                    p;
for (p=&state[0];p<&state[n];p++){              /*initialize state*/
p->predecessor=-1
p->length=INFINITY
p->label=tentative;
}
state[t].length=0; state[t].label=permanent ;
k=t ;                                            /*k is the initial working node */
do{                                              /* is  the better path from k? */
for I=0; I<n; I++)                               /*this graph has n nodes */
if (dist[k][I] !=0 && state[I].label==tentative){
          if (state[k].length+dist[k][I]<state[I].length){
             state[I].predecessor=k;
             state[I].length=state[k].length + dist[k][I]
           }
}
/* Find the tentatively labeled node with the smallest label. */
k=0;min=INFINITY;
for (I=0;I<n;I++)
      if(state[I].label==tentative && state[I].length <
min)=state[I].length;
     k=I;
   }
state[k].label=permanent
}while (k!=s);
/*Copy the path into output array*/
I=0;k=0
Do{path[I++]=k;k=state[k].predecessor;} while (k>=0);
}
```

Fig.6 Example of Dijkstra algorithm in C code

CONCLUSIONS

Some general conclusions that can be drawn from this study are summarized below.

1. Providing multiple routes is beneficial in improving the quantity of service.

2. Oscillations in traffic load must be avoided but sensitivity to congestion may be significant.

3. The failure of the traffic management center may be dangerous to a centralized routing management system.

4. Adaptive routing with constantly updated information is helpful in avoiding congested routes.

5. Quality of service, quantity of service and speed are the three most important performance measures for any routing algorithm.

REFERENCES

[1] Ash, G. Dynamic Routing in Telecommunications Networks, McGraw Hill, NY, 1998.
[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. ftp://ftp.cs.arizona.edu/xkernel/Papers/jsac.ps.Z
[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 Magazin*e, 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, http://www.aciri.org/floyd/papers/report_Jan01.pdf
[10] Fortz,B., J.Rexford, M.Thorup. Traffic Engineering with Traditional IP Routing Protocols. *IEEE Communication Magazine,* 2002, http://www.research.att.com/~jrex/papers/ieeecomm02.ps
[11] S. Halabi, S., Internet Routing Architectures, 2nd Ed ., Cisco Press, 2000.
[12] Huitema, C. Routing in the Internet. Englewood Cliffs, New Jersey, Prentice Hall, 1995
[13] P. Gupta, P., N. McKeown, Algorithms for Packet Classification, *IEEE Network Magazine,* Vol. 15, No. 2 (Mar./Apr. 2001), pp. 24–32, http://klamath.stanford.edu/~pankaj/paps/ieeenetwork_tut_01.pdf

ABOUT THE AUTHORS

Prof. Todor Stoilov, D.Sc., PhD, Institute of Computer and Communication Systems, Bulgarian Academy of Sciences, Phone: +359 2 73 78 20, E-mail: todor@hsi.iccs.bas.bg

Assoc. Prof. Krasimira Stoilova, PhD, Institute of Computer and Communication Systems, Bulgarian Academy of Sciences, Phone: +359 2 979 27 74,
E-mail: k.stoilova@hsi.iccs.bas.bg

This research is partly supported by the National Scientific Fund of Bulgaria, project № ВУ-966, 2005.